NASA CR-144891

STICAP

A LINEAR CIRCUIT ANALYSIS PROGRAM WITH STIFF SYSTEMS CAPABILITY

Volume III - Systems Programmer's Manual

by Charlie H. Cooke and M Niel Ransom

(NASA-CR-144891) STICAP: A LINEAR CIRCUIT ANALYSIS PROGRAM WITH STIFF SYSTEMS CAPABILITY. VOLUME 3: SYSTEMS (Old Dominion Univ., Norfolk, Va.) 126 p HC $6.00 CSCL 09B G3/61

N76-13800

Unclas
05365

Prepared under Contract NAS1-9434, T25 by
OLD DOMINION UNIVERSITY
School of Engineering
Norfolk, VA 23508

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

STICAP (Stiff Circuit Analysis Program) is a FORTRAN IV, Version 2.3, computer program written for the CDC-6400-6600 computer series and SCOPE 3.0 operating system. It provides the circuit analyst a tool for automatically computing the transient responses and frequency responses of large linear time invariant networks, both stiff and non-stiff. The circuit description and user's program input language is engineer-oriented, making simple the task of using the program.

Three volumes of documentation are available for the STICAP program; a theory manual, a user's manual, and a system's programmers manual. Volume I describes the engineering theories underlying STICAP and gives further references to the literature. Volume II, the user's manual, explains user interaction with the program and gives results of typical circuit design applications. Volume III depicts the program structure from a system's programmers viewpoint and contains flow charts and other software documentation.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

This document contains material describing the internal program structure of the STICAP program. An overview of the composite system structure is given. The overlay structure is described, and the routines composing each overlay are included, for each routine generated under contract NAS1-9434-25. A description of the CORNAP and Gear routines may be found in the references [1] and [2] below, which contain as much documentation as is available for these programs.

The program is written using CDC FORTRAN 4, Version 2.3. It is machine compatible with the CDC 6400-6600 computer series, and runs under the SCOPE 3.0 operating system. It is segmented in overlays of 70K or less using the SCOPE OVERLAY capability. All I/O is accomplished using standard I/O files. I/O files are equivalenced so that File 5 is used for input and File 6 for output. No other files are used by this program.

---

1. Pottle, C., "Program Cornap - FORTRAN Computer Routine," System Theory Group, School of Electrical Engineering, Cornell University, Ithaca, N. Y., 1968.

2. Gear, C. W., Algorithm 407 - DIFSUB for Solution of Ordinary Differential Equations [D2], Communications of the ACM, March, 1971, Vol. 14, No. 3, 185-190.

CHAPTER II

OVERLAY STRUCTURE

The first component of the program overlay structure (see Figure 1) consists of a root segment (a main overlay) containing a supervisor routine STICAP, a commonly used routine USERCN, and blocks of labeled and unlabeled common core storage, also used by many routines. The supervisor routine obtains control at initial execution and coordinates the calling into core of the successive overlays. This is accomplished by means of subsequent primary overlays, of which each successive one replaces the preceding, with the number called dependent upon the mode of operation selected by the user.

The first primary overlay consists of a block of routines which read the input circuit description cards, scaling card, and mode card. The circuit description cards, whose format has been altered from that of CORNAP, provide the user an easy means of describing the circuit. The scaling card, optionally present, determines the way the circuit equations are to be scaled for numerical computation of a solution. The mode card determines the program mode, or method by which the solution is to be obtained.

The second primary overlay consists of the original program CORNAP, minus the data input processing routines, whose function has been accomplished by the routines mentioned above. The routines of this overlay process the circuit description data previously read, to obtain the mathematical equations governing the circuit. If the solution is to be obtained by the CORNAP mode, the original CORNAP routines continue processing the circuit equations.

In the event that either the Gear or Matrix mode is selected, the third primary overlay is called into core. Data compiled by

Root Segment
STICAP
USEFCN
LABELED COMMON
BLANK COMMON

Primary Overlays

1	2	3		MATRIX		
FEED	C	COMPAK	INTGR			
INPUT	O	APACK*	DIFSUB**	SETSVE	INTCL	OUTPNT
MODE	R	CROLCR	DATPNT	VAN[illegible]V	SVECOS	CLSOL
APACK	N	SRCORD	DIFFUN	EIGNV	SVESIN	TABPNT
ATODEC	A	RUNCTL	PEDERV	HESFT	SVESTP	ATAN3
NAME	P	ATODEC*	DECOMP**	ORTRAN	INIVAL	
NODE		DECEQV*	SOLVE**		MPROD	
DECEQV		CROLMX	4th Primary overlay optionally chosen		EAT	
OUTPUT		SRCINT			SCEYAL	
SFSCAL		RUNCRL			TRANS	
		CROLCM			STEADY	
					STORE	

Secondary Overlays

Figure 1 - Overlay Structure

*Appears in a previous overlay also.

**Gear's ALGORITHM 407 Routines.

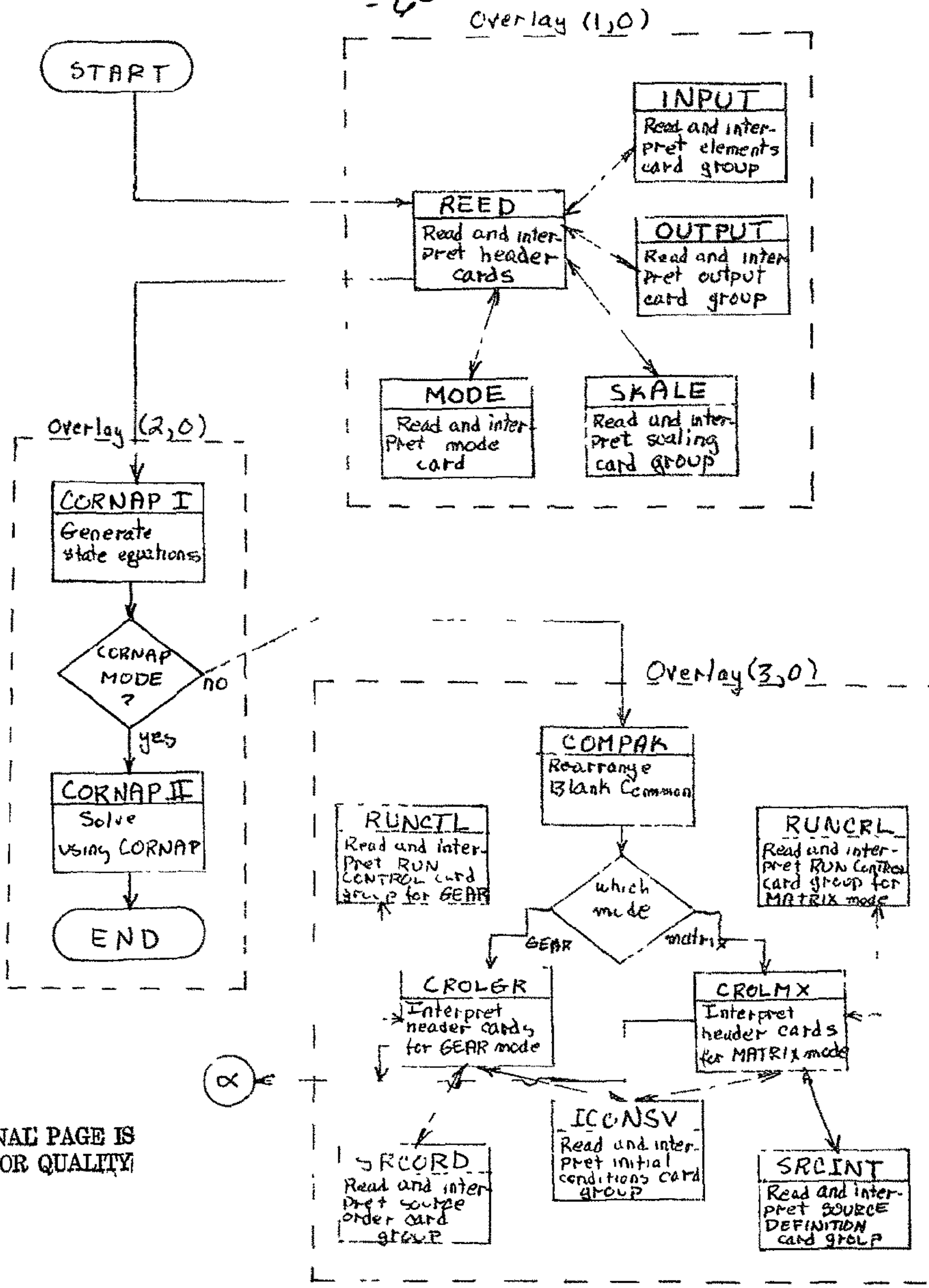
CORNAP is compacted into a more economical form by the routine COMPAK. The control cards and data input cards for the mode selected are processed, and the final primary overlay is called.

The final overlay routines are determined by the mode selection. In one case the routines of Gear are used for numerical integration; in the other the matrix technique is used to obtain the solution.

CHAPTER III

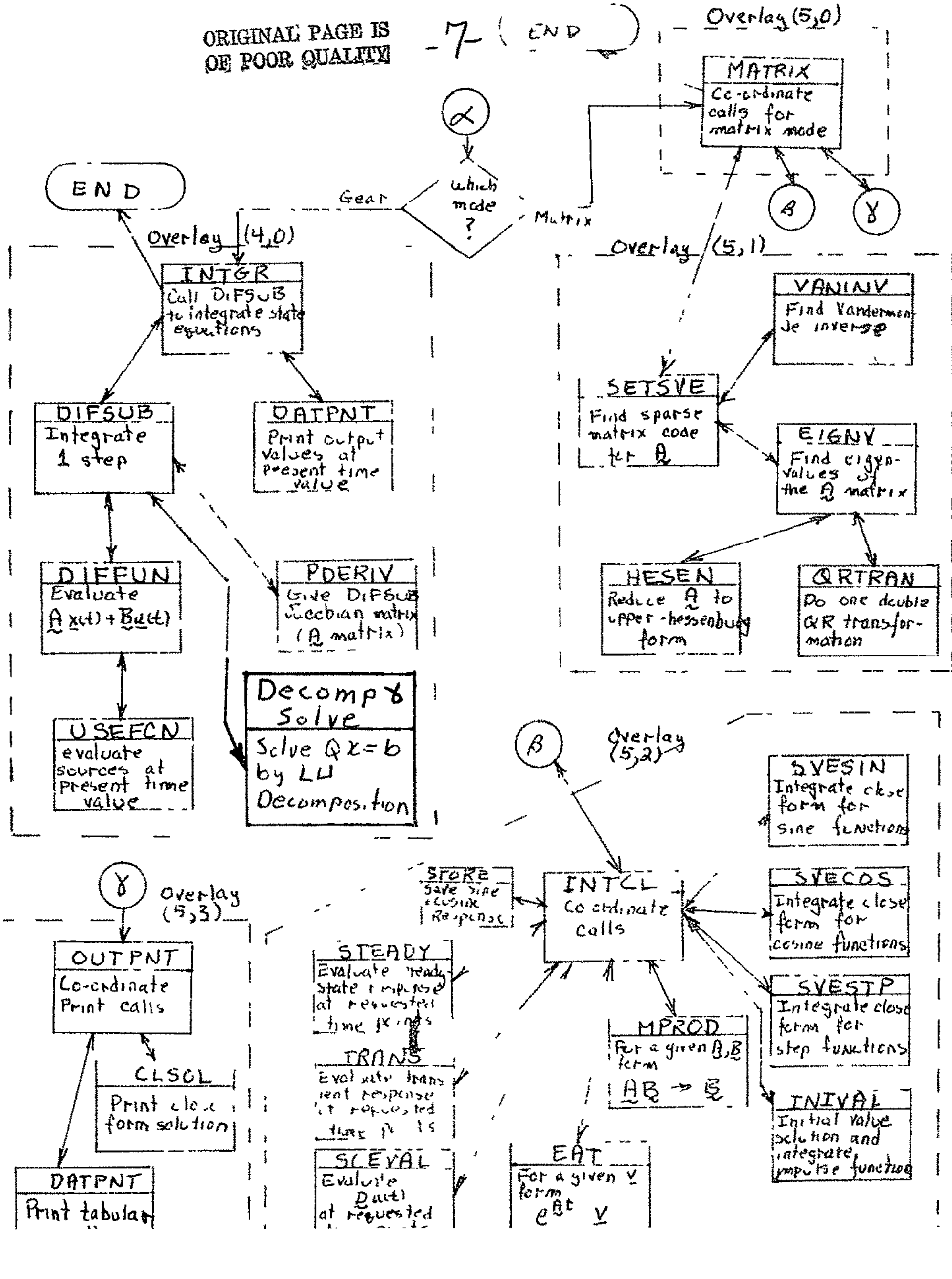
SYSTEM OVERVIEW

An overview of the STICAP system indicating the overall logical flow of the program appears in the following pages. The figures indicate the routines within each overlay; the functions performed; and the calling sequences, or flow of control.


Overlay (1,0)
START
INPUT
Read and interpret elements card group
REED
Read and interpret header cards
OUTPUT
Read and interpret output card group
MODE
Read and interpret mode card
SKALE
Read and interpret scaling card group
Overlay (2,0)
CORNAP I
Generate state equations
CORNAP MODE ?
no
yes
CORNAP II
Solve using CORNAP
END
Overlay (3,0)
COMPAK
Rearrange Blank Common
RUNCTL
Read and interpret RUN CONTROL card group for GEAR
RUNCRL
Read and interpret RUN Control card group for MATRIX mode
which mode
GEAR
matrix
CROLGR
Interpret header cards for GEAR mode
CROLMX
Interpret header cards for MATRIX mode
α
SRCORD
Read and interpret source order card group
ICONSV
Read and interpret initial conditions card group
SRCINT
Read and interpret SOURCE DEFINITION card group

ORIGINAL PAGE IS OF POOR QUALITY

- 7 - (END)



CHAPTER IV

FUNCTIONS OF THE INDIVIDUAL ROUTINES

In this section a brief description of the function performed by each contractor supplied* routine is given, and the routines are categorized by the overlay in which they operate. We remark that occasionally a routine may be used in several overlays; hence it may appear more than once. For a more complete description of a routine, the next section may be consulted.

The name and function of the various routines will now be given.

I. Root Segment Routines

1. USEFCN - Supplies the user a means of specifying the independent sources of the circuit.
2. STICAP - Supervisor routine which coordinates the calling into core of the primary overlays.

II. First Primary Overlay

1. REED - Processes title card and scaling card; coordinates routines which read CORNAP circuit description.
2. INPUT - Reads circuit description cards in a free format.
3. MODE - Reads Mode select card and processes it.
4. APACK - Packs into one word the first character of each word from an array of words.
5. ATODEC - Interprets free format floating point numbers.
6. NAME - Scan routine for breaking out a name from a data field of free form data.

---

*Some routines not contractor supplied are also indicated. Such routines are Langley routines or else Gear supplementary routines.

7. NODE - Scan routine for breaking out an integer number from a data field of free form data.
8. DECEQV - Changes alphanumeric BCD characters into internal integer format on the CDC - 6600 computer.
9. OUTPUT - Reads and processes cards indicating the circuit outputs specified by the user.
10. ZFSCAL - Reads scaling cards and stores scaling parameters.

III. Second Primary Overlay

CORNAP - The body of routines composing the original program CORNAP, minus routines which read into core the circuit description cards. This overlay also contains the logical decisions required for carrying out the various print options, such as state equations, transfer functions, etc.

IV. Third Primary Overlay

1. APACK - Previously described; first primary overlay.
2. CROLGR - Coordinates the subroutines which read the source order; initial conditions, run controls card group, and end card for the Gear mode.
3. ICONSV - Reads and processes the initial conditions card groups for either the Gear or Matrix modes.
4. SRCORD - Reads and processes the source order card groups for the Gear modes.
5. RUNCTL - Reads and processes run control cards for Gear mode.
6. ATODEC - Previously described; first primary overlay.

7. DECEQV - Previously described: first primary overlay.
8. CROLMX - Coordinates the subroutines which read the source definition, initial conditions, run controls card groups, and the END card for the Matrix mode.
9. SRCINT - Interprets the source definition card group for the Matrix mode.
10. RUNCRL - Reads and processes run controls card group for the Matrix mode.
11. COMPAK - Rearranges data tables output by CORMAP circuit translation routines into a more economical form.
12. CROLGM - Controlling routine for third overlay.

V. <u>Gear Primary Overlay</u>

1. INTGR - Coordinates Gear's routine DIFSUB for numerical integration and determines when the stop condition is reached.
2. DIFSUB - Integrates the circuit equations in the Gear mode.
3. DATPNT - Prints output data.
4. DIFFUN - Evaluates the right members of the state variable differential equations of the circuit, given a specified time point.
5. PEDERV - Evaluates the Jacobian Matrix of the system equation for the circuit.
6. DECOMP - LU decomposition routine used in tandem with SOLVE to solve a linear system.
7. SOLVE - Second stage of a LU decomposition linear equation solver pair of routines, used by DIFSUB.

VI. Matrix Primary Overlay

1. MATRIX - Calls matrix mode secondary overlays.
2. SETSVE - A controlling routine for matrix mode, which also obtains a sparse matrix code.
3. VANINV - Inverts a Vandermonde matrix.
4. EIGNV - Obtains system eigenvalues using QR transformation.
5. HESEN - Reduces a matrix to upper Hessenburg form.
6. QRTRAN - Performs a single double-step of the QR transformation, avoiding complex algebra.
7. INTCL - Controls the closed form integration performed by the matrix mode.
8. SVECOS - Processes integration of COSINE inputs.
9. SVESIN - Processes integration of SINE inputs.
10. SVESTP - Processes integration of STEP inputs.
11. INIVAL - Integrates for impulse driving functions or other driving functions with a specified initial value.
12. MPROD - Forms a matrix product.
13. EAT - Calculates the matrix exponential exp(AT) and multiplies into a vector.
14. SCEVAL - Evaluates the $D\bar{u}$ vector, $\bar{u}$ the vector of independent source inputs, $\bar{y} = C\bar{x} + D\bar{u}$ the requested output vector, at a specified time.
15. TRANS - Evaluates transient closed form solution at a specified time point.
16. STEADY - Evaluates steady state closed form solution at a specified time point.
17. STORE - Stores symbolic steady state closed form solution.

18. OUTPNT - Controls printing of matrix mode outputs.

19. CLSOL - Prints symbolic closed form solutions.

20. TABPNT - Prints tabular form of matrix mode solutions.

21. ATAN3 - Combines a sine and cosine function to obtain a single sine function.

CHAPTER V

INDIVIDUAL ROUTINE DESCRIPTIONS

13

Subroutine: USEFCN

This routine allows the user the capability of defining the form of the input vector for state-input and state output equations of the network, for the Gear mode:

$$\frac{dx}{dt} = A\bar{x} + B\bar{u}$$

where

$\bar{x}$ is the state vector of the circuit,

$\bar{u}$ is the vector if independent sources,

$\bar{y}$ is the vector of outputs requested by the user.

The user must either furnish FORTRAN equations defining the independent sources, or else must supply a routine for computing their values, at a given time. The manner in which this may be accomplished is described in the user's manual.

The net effect of an entry to USEFCN must be the following: Given an initial time T and a vector array U, compute values of the independent sources at time T and store as the components of the vector U. The order in which these components are stored must correspond to the order specified by the user when assembling the source order cards group (see user's manual).

Inputs: a value of T and a vector array U

Outputs. The values of the independent sources at time T, stored in the array U, in the order specified by the user's source order cards.

Program IEED

This program reads the title card and calls INPUT, OUTPUT, and ZFSCAL to read the elements, output, and scaling card groups. The card groups are signaled by the following header cards:

*ELEMENTS

*OUTPUTS

*SCALING

where the asterisks appear in column one. The element and outputs card groups must always appear. The scaling card group need be present only if the circuit is to be scaled. The outputs card group must not be given before the elements card group.
This program then calls MODE to interpret the mode card. The mode card must appear after the above card groups.
The outputs from this routine are described in the comment cards preceding the subroutines INPUT, OUTPUT, and ZFSCAL.

SUBROUTINE INPUT

This subroutine reads the elements card group. All cards read until the next card with an asterisk in column 1 are considered element cards.
The cards are of the form

NAME N1 N2 VALUE

for passive elements or independent sources and

NAME N1 N2 VALUE*NAME2

for dependent sources. NAME,N1,N2, and VALUE must be separated by at least one blank. The asterisk between VALUE and NAME2 need not be present. NAME is the user name for the circuit element. This name must be no more than 4 characters long. The first letter of the element name tells the type of element as follows.

First Letter	Element Type
V	Voltage Source
I	Current Source
R	Resistor
L	Inductor
C	Capacitor
K	Coefficient of Coupling
M	Mutual Inductance or Capacitance

N1 and N2 for non-mutual elements are the number of the nodes between which the element is connected. The circuit nodes should be numbered compactly from zero, although the failure to do so is non-fatal. N1 is given a positive reference with respect to N2. For mutual elements N1 and N2 are the name of the two elements involved. For dependent sources NAME2 is the name of the controlling element with a V or an I added to the beginning of the name depending on whether the source is voltage or current controlled.

A feature not normally employed by the average user is the following:

A capacitor may be forced into the -proper tree- by placing the word tree after the description of the element.

NAME N1 N2 VALUE TREE

An inductor may be forced into the cotree by similarly placing the word cotree after the element description

NAME N1 N2 VALUE COTREE

OUTPUT PARAMETERS

LIST A 3 dimensional matrix whose I,J,K element tells information J of the Ith element of type K.

K	Type
1	Dependent Voltage Source
2	Independent Voltage Source
3	Zero Valued Resistor or Inductor
4	Tree Forced Capacitor
5	Capacitor
6	Resistor
7	Inductor
8	Cotree Forced Inductor
9	Zero Valued Capacitor
10	Independent Current Source
11	Dependent Current Source
12	Mutual Element

J	Information
1	User Name for Element
2	Element Value
3	Zero (later used to define outputs)
4-5	For non-mutual elements the nodes between which the element is placed
4-5	For mutual elements the names of the two elements involved

NF A vector whose Ith element is the number of elements of type I (defined as in LIST).

CONBCH An array having two columns whose Ith row is the name of the element which controls the Ith dependent source. Column 1 is for dependent voltage sources and column 2 is for dependent current sources.

ICON An array constructed like CONBCH, but whose elements tell the type of control of the dependent sources. 1 for voltage controlled and 2 for current controlled.

NODEG A vector whose Ith element tells the number of elements and sources connected to the Ith node.

NEX The number of nodes to which no elements are connected.

NODES The largest numbered node given by the user.

Subroutine: MODE(IMODE,IPRINT)

Description: This subroutine reads the mode card to determine what program mode is selected and what print options are chosen.

Program execution will be terminated if an error occurs in describing the program mode. An error in describing a print option is non-fatal, but the corresponding print request will be ignored.

Input Parameters: Mode Card - This card will be in the following form

*MODE NAME, OPTION 1, OPTION 2, OPTION 3

The card is in free form, except for an asterisk in column one. The absence of the asterisk is non-fatal. MODE NAME is the name of the solution program and is one of the following

GEAR

CORNAP

MATRIX.

These names may be abbreviated or misspelled as long as the first letter of the first word is correct.

The options following the name of the solution program are print requests. The are separated by commas and are of the following form:

SOLUTION EQUATIONS

STATE EQUATIONS

TRANSFER FUNCTIONS

These names may be abbreviated or misspelled, as long as the first two letters are correct. The order in which the print options appear is insignificant, nor do all or any need be present.

<u>Output Parameters</u>: IMODE - An integer telling what solution program to use

IMODE = 1 CORNAP

IMODE = 2 GEAR

IMODE = 3 MATRIX

IPRINT - A vector telling what print options were requested.

IPRINT(I) is 1 if option I was requested and 0 if it was not. The elements of IPRINT relate the following:

IPRINT(1) State equations

IPRINT(2) Transfer functions

IPRINT(3) Solution equations

<u>Additional Subroutines Required</u>

APACK

<u>Examples of Usage</u>

1. Card read:

*GEAR MODE

The subroutine would return

IMODE = 2

IPRINT(1) = 0

IPRINT(2) = 0

IPRINT(3) = 0

2. Card read

*MATRIX, SOLUTION EQUATIONS, TRANSFER FUNCTIONS

The subroutine would return

IMODE = 3

IPRINT(1) = 0

IPRINT(2) = 1

IPRINT(3) = 1

3. Card read

CORNAP, TRANSFER FUNCTIONS

Here the asterisk in column 1 was omitted. The subroutine would return

IMODE = 1

IPRINT(1) = 0

IPRINT(2) = 1

IPRINT(3) = 0

4. Card read

*C,ST EQU,TR

Here the names of the solution program and the print requests were abbreviated. The subroutine would return

IMODE = 1

IPRINT(1) = 1

IPRINT(2) = 1

IPRINT(3) = 0

Subroutine: APACK(NLET,ARRAY,NWORD)

Purpose: To pack the first character of an array of words into one word.

Input Parameters: ARRAY - An array of words containing alpha-numeric BCD characters.

NLET - The number of words in ARRAY

Output Parameters: NWORD - A word containing the first character of the first word in ARRAY as its first character, the first character of the second word in ARRAY as its second character, and so on until ARRAY is exhausted. The remainder of NWORD is filled with blanks.

Machine Dependancy: This subroutine assumes 60 bit words with 6 bit alpha-numeric characters stored from the left.

Example of Usage:

```
ARRAY(1) = 3HABC
ARRAY(2) = 1HD
ARRAY(3) = 4HEFGH
CALL APACK(3,ARRAY,NWORD)
```

Upon return from the subroutine call, NWORD would contain 3HADE.

SubroutineSRequired:

RSHFT
RSHFTA
system subroutines; not STICAP routines.

Subroutine: ATODEC (NUFFER, LSRT, LSTP, DEC, NERR)

Description: This subroutine will interpret a number which was read from a card and stored in an array in alpha-numeric BCD characters.

The number may be in integer, decimal or exponential form. The numbers and its exponent are presumed positive if no sign is given. Numbers in exponential form need not contain a decimal point.

All blanks within the input field are ignored.

The subroutine starts at some given point in the input field and interprets until the first character not part of the number is found. This does not necessarily mean a non-numeric character. The location of this character and the number found is outputted by the subroutine. If no number is found before a non-numeric character, such as a digit or a + or - sign, the number being interpreted is assumed to be a coefficient of 1.0. If no number is found before the end of the input field, a 0.0 is given by the subroutine as the number.

Limitations: The magnitude of the number found must be between 10^{-290} and 10^{+290} and contain no greater than 15 digits.

Input Parameters:

NUFFER - A vector containing the card read in 80A1 format.

LSRT - A starting location in NUFFER where interpreting is to begin.

Output Parameters:

DEC - The number found.

LSTP - The location in NUFFER of the next non-blank character following the number found. If the rest of the card was blank, LSTP = 81

NERR - Error Code

NERR = 0 No error present

NERR = 1   Number is more than 15 digits long

NERR = 2   Number is not between 10^{-290} and 10^{+290} in magnitude.

<u>Machine Dependancy:</u> There is no machine dependancy within the subroutine itself, however a function which it calls, DECEQV, is machine dependent.

<u>Examples of Usage:</u>

```
      READ 10, (NUFFER(I),I=1, 80)
   10 FORMAT (80 A1)
```

(1) Card contains   bbb2.147bbb...

```
      CALL ATODEC(NUFFER,1,LSTP, DEC,NERR)
```

After the call

NUFFER remains unchanged

LSTP   contains 81

DEC   contains 2.147

NERR   contains 0

(2) Card contains   bbA=2SIN(4T)bb...

```
      CALL ATODEC(NUFFER,5,LSTP,DEC,NERR)
```

After the call

NUFFER remains unchanged

LSTP   contains 6

DEC   contains 2.0

NERR   contains 0

(3) Card contains   bb216E-3EXP(-X)bb...

```
      CALL ATODEC(NUFFER,1,LSTP,DEC,NERR)
```

After the call

NUFFER remains unchanged

LSTP contains 9

DEC contains .216

NERR contains 0

(4) Card contains -SIN3Xbbb

CALL ATODEC(NUFFER,1,LSTP,DEC,NEPR)

After the call

NUFFER remains unchanged

LSTP contains 2

DEC contains -1.0

NERR contains 0

(5) Card contains bb3.14159265358979 32bb...

CALL ATODEC(NUFFER,1,LSTP,DEC,NERR)

After the call

NUFFER renains unchanged

LSTP undefined

DEC undefined

NERR contains 1

(6) Card contains bb-2.14E-5bb21SIN(X)bbb...

CALL ATODEC(NUFFER,1, LSTP, DEC,NERR)

After the call

NUFFER remains unchanged

LSTP undefined

DEC contains -2.14

NERR contains 2

Subroutines required:

NCODE, DECEOV

Subroutine:

NAME(NUFFER,LSRT,LSTR,NERR)

This subroutine scans a card image to find an element name. This name must be less than 4 characters long.

Input Parameters:

NUFFER	The card image in 80A1 format.
LSRT	Starting location where scanning is to begin

Output Parameters:

NWORD	The element name found.
LSTP	Location of blank after the element name.
NERR	Error code.
	NERR = 0 No error present.
	NERR = 1 Name over 4 characters long.
	NERR = 2 No name present (rest of card blank).

Additional Subroutines Required:

APACK

Subroutine:

NODE(NUFFER,LSRT,LSTP,NUM,NERR)

This subroutine scans a card image from a given starting location to find the node number. The subroutine requires this node number to be followed by a blank. The largest node number must be less than 64.

Input Parameters:

NUFFER	An array containing the card image in 80A1 format.
LSRT	The starting location where scanning is to begin.

Output Parameters:

NUM	The node number found.
LSTP	The location of the first blank following the node number.
NERR	Error code
	NERR = 0 No error present
	NERR = 1 A blank does not follow the number.
	NERR = 2 Node number greater than 64.
	NERR = 3 Card after LSRT blank.

Function: DECEQV (NCOEF, LCNT)

Function Purpose: Convert an integer whose digits are BCD characters to an equivalent floating point number.

Limitations: The integer being converted must be less than 15 digits long.

Input Parameters:

NCOEF - An integer array whose Ith element is the Ith digit of the integer being converted, read from the left in A format.

LCNT - The number of digits in the number being converted.

Output Parameters:

DECEQV - The equivalent floating point number.

Machine Dependency: This function assumes 60 bit words with alphanumeric characters in CDC 6000 series BCD code stored on the left.

Example of Usage:

```
NCOEF (1) = 1H2
NCOEF (2) = 1H8
NCOEF (3) = 1H4
NCOEF (4) = 1H5
DEC = DECEQV (NCOEF, 4)
```

Upon return from the subroutine call DEC would contain 2845.0.

Subroutines Required:

RSHFT (a system subroutine; not a STICAP routine)

Subroutine: OUTPUT

This subroutine reads the output card group to determine the element currents and voltages selected as circuit outputs. This subroutine will terminate circuit processing if no outputs are selected or if all outputs contain errors.

All cards read until a card with an asterisk in column 1 is found are considered output cards. The cards are of the form

VNAME

or

INAME

depending on whether the voltage across or the current through NAME is desired. Here NAME is the name of a circuit element given in the elements card group.

Input Parameters:

NF - A vector of length 12 whose elements tell the number of each of the 12 types of elements contained in the circuit.

LIST - A matrix whose (I,1,J) element tells the name of the Ith element of type J.

Output Parameters:

LIST - A matrix whose (I,4,J) element gives the output information for the Ith element of type J.

LIST (I,3,J)=0 No output taken from this element.

LIST (I,3,J)=1 Voltage across this element is an output.

LIST (I,3,J)=2 Current through this element is an output.

Subroutine: ZFSCAL

This subroutine reads the scaling card group. The circuit will be scaled around a frequency or impedance value if one of the following cards are present

FREQUENCY = A

IMPEDANCE = A

where A is a floating point number in free format. Scaling requests containing errors are ignored.

Input Parameters: None

Output Parameters:

ZSCALE - The impedance around which the circuit is to be scaled.

FSCALE - The frequency around which the circuit is to be scaled.

NUFFER - A vector containing the next header card.

Subroutine: COMPAK (EPS,KA,KSC,NDERIV,NROWA,NLC,IMODE,IPRINT,FSCALE, ZSCALE,NAMLC)

Description: This subroutine takes information assembled by CORNAP subroutines and puts it in the form used by the GEAR and MATRIX mode subroutines.

This subroutine will terminate execution whenever derivatives of the inputs are required by the output equations or whenever derivatives of the inputs occur higher than the first degree.

Whenever the only derivatives of the inputs appear in the state variable equations and then only the first degree, the state variable and output equations will be of the form

$$\underline{\dot{X}} = \underset{\sim}{A}\underline{X} + \underset{\sim}{B}\underline{u} + \underset{\sim}{B}_1\underline{\dot{u}}$$

$$\underline{Y} = \underset{\sim}{C}\underline{X} + \underset{\sim}{D}\underline{u}$$

The following substitutions are then made

$$\underset{\sim}{A} \cdot \underline{B1} + \underset{\sim}{B} \rightarrow \underset{\sim}{B}$$

$$\underset{\sim}{C} \cdot \underline{B1} + \underset{\sim}{D} \rightarrow \underset{\sim}{D}$$

The new circuit equations now take the form

$$\dot{\underset{\sim}{q}} = \underset{\sim}{A}\underset{\sim}{q} + \underset{\sim}{B}\underset{\sim}{u}$$

$$\underset{\sim}{Y} = \underset{\sim}{C}\underset{\sim}{q} + \underset{\sim}{D}\underset{\sim}{u}.$$

The names of the circuit outputs are augmented by adding an I or V in front of the old name depending on whether the current through this element or the voltage across this element is the output.

Inputs and Outputs: All inputs are brought in by the two arrays BLOCK and NAMES stored in common and by the argument list of the subroutine.

The following variables are inputs to the subroutine but are not outputted.

KA,KSC,NROWA - Constants used to locate the A, B, B1, C, D, and D1 matrices in BLOCK

D1 - Coefficient matrix of source derivatives in the output equations

The storage location of the remaining input and output variables are given in Figure 1.

Machine Dependency: The process of adding an I or V in front of the output names is machine dependent.

The addition of the I or V is a means of more readily identifying current and voltage variables for printout and/or scaling purposes.

FIGURE 1

Variable	Length	Starting Location When Used	Output Starting Location in COMAPY	Meaning
A	900	BLOCK(KA+NO,$\frac{NROWA}{2}$,KSC)	1	System Matrix
B	300	BLOCK(KA+NO,$\frac{NROWA}{2}$+KSC+NSTV)	901	Source Coef. Matrix in State eqns
C	300	BLOCK(KA,$\frac{NROWA}{2}$ + KSC)	1201	State Variable Coef Matrix in Output eqns
D	100	BLOCK(KA $\frac{NROWA}{2}$+KSC+NSTV)	1501	Source Coef Matrix in Output eqns
P1	300	BLOCK(KA+NO,$\frac{NROWA}{KSC}$+NSTV KSC+NIN)	1601	Coef Matrix of Source Deriv. in State eqns
NSTV	1	NAMES (17)	1901	Number of State Variables
NIN	1	NAMES (17)	1902	Number of inputs
NO	1	NAMES (17)	1903	Number of outputs
NLC	1	Subroutine argument	1904	Number of coils and capacitors
NDERIV	1	Subroutine argument	1905	Number of source derivatives
IMODE	1	Subroutine argument	1906	Solution Method, 2-GEAR, 3-MATRIX
IPRINT	3	Subroutine argument	1907	Extra Printouts desired
NAMIN	10	NAMES (49)	1910	Names of inputs
NAMSV	30	NAMES (33)	1920	Names of state variables
NAMOUT	10	NAMES (65)	1950	Names of outputs
NAMLC	30	Subroutine argument	1960	Names of coils and capacitors
FSCALE	1	Subroutine argument	1990	Frequency scale factor
ZSCALE	1	Subroutine argument	1991	Impedance scale factor
EPSLN	1	Subroutine argument	1992	Smallest non-zero matrix element

ORIGINAL PAGE IS OF POOR QUALITY

30

Subroutine: CROLGR

Description: This subroutine calls the subroutines to read the source ordering, initial conditions, and run controls card groups and the end card. These card groups are signaled by header cards which contain asterisks in their first column. The names on these header cards and the expected order in which they appear are:

* SOURCE ORDERING

* INITIAL CONDITIONS

* RUN CONTROLS

* END

The source ordering, initial conditions, and run controls card groups need not appear in this order but must appear before the end card. In this order, however, they will process faster.

The source order card group must appear if more than one source is defined in the circuit description. Failure to do this will result in termination of program execution.

If the run controls card group does not appear, the following run controls are assumed:

ISTOP = 1

TO = 0.0

TSTART = 0.0

H = 10^{-4}

HMIN = 0.0

METHOD = 1

IHMAX = 1

MAXPTS = 100

INCRIT = 3

A blank header card will be assumed to be the end card. If the end card does not appear, the end of file card is assumed to be the end card.

An unrecognizable header card will result in termination of program execution.

Usage: This subroutine is called by the main program when the Gear option has been chosen, immediately after control is returned from CORNAP.

Routine Name: ICONSV

Function: Read cards containing the names and initial values of the components of the initial state vector. Print all nonzero conditions used by the program and/or any diagnostic error messages generated.

Setting: At execution time the '* INITIAL CONDITIONS" card has just been processed, and the routine expects to read and process cards containing the initial circuit conditions, stopping at the first card read which contains an * in column 1.

Inputs:

Initial condition cards

NSTV - The number of state variables CORNAP obtains

NPSTV - The number of excess state variables*

NAMSV - Names of CORNAP obtained state variables

NAMLC - Names of excess state variables

Outputs: Error diagnostics and/or state vector of initial conditions. Components improperly formatted are set to zero. Excess components are ignored.

---

*An excess state variable is a capacitor voltage or inductor current whose value is dependent upon other capacitor voltages or inductor currents.

Subroutine: SRCORD

Function: This subroutine reads the source order cards to determine a correspondence between the way the sources are ordered in the state variable equations and the order in which they are computed and stored by USEFCN.

Inputs:

NIN - The number of independent sources defined in the circuit description.

NAMES - A vector containing the user names of the independent sources in the same order they appear in the state variable equations.

Outputs:

NORDER - A vector whose Ith element is the position in the Vector U(I) of USEFCN which corresponds to the position of the Ith source in the state variable equation as ordered by the circuit translation routines.

NUFFER - A vector containing the next control card.

Error messages are printed and the program terminated for the following conditions:

(1) An input is specified more than once.

(2) An input specified is not found in "NAMES."

(3) Not all inputs defined in the circuit description are specified on the source order cards.

Additional inputs: This subroutine will read the source order cards. These cards contain the names of the sources, one per card in free format, in the order they are defined in USEFCN. All cards read are assumed to be source order cards, unless they contain an asterisk in column 1. When a card containing an asterisk in column 1 is read, it is stored in NUFFER and the source order cards are assumed to be complete.

<u>Subroutine:</u> RUNCTL

<u>Description</u>. This subroutine reads the run controls card group for the Gear mode. The completion of this card group is signaled by a card with an asterisk in column 1 or the end of file card.

<u>Input Parameters</u>: The run controls cards must be one of the following:

INITIAL TIME = A

STEP RESPONSE

IMPULSE RESPONSE

PRINT START = A

STOP TIME = A

POINTS PRINTED = N

EPS = A

ADAMS INTEGRATION

HMAX = A

HMIN = A

HINIT = A

Here A is a floating point number and N is an integer. A description of the information conveyed by such cards is contained in the user's manual for STICAP.

<u>Output Parameters</u>:

TO - Initial value of time

TSTART - Time at which output printings is to begin

ISTOP - Stop condition

ISTOP = 1 Maximum number of points printed

ISTOP = 2 Maximum value of time

MAXPTS - Maximum number of points printed

TSTOP - Maximum value of time

H - Initial step size

EPS - Error control

METHOD - Integration method

METHOD = 0 Adams method

METHOD = 1 Stiff integration method

INCRMT - Number of integration steps between output printings

MAXDER - Maximum derivative used by the integration method.

MAXDER is set to 7 for METHOD = 1 and 8 for METHOD = 0.

HMAX - Maximum integration step size allowed

HMIN - Minimum integration step size allowed

NPUT - Type of input

NPUT = 0 User defined

NPUT = 1 Impulse

NPUT = 2 Step

NUFFER - Vector containing the next header card. If an end of file card was encountered, NUFFER(2) is set to 1HE which will be interpreted as the end card.

Default Options: Unless specified by the run controls cards, the following values are assumed.

ISTOP = 1

TO = 0.0

TSTART = TO

NPUT = 0

HMAX = (TSTOP-TO)/10 if TSTOP was specified.

HMIN = 0.0

HMAX = 0.0 (interpreted as unbounded)

MAXPTS = 100

EPS = 10^{-4}

METHOD = 1

INCRMT = 3

H = 10^{-4}

MAXDER = 7

Machine Dependence: This subroutine itself is machine independent but calls the machine dependent routines APACK and ATODEC.

Subroutines Required: APACK, ATODEC

Program: CROLMY

This subroutine calls the subroutines to read the source definition, initial conditions, and run controls card groups and the END card. These card groups are signaled by header cards which contain asterisks in their first columns. The names on these header cards are

* SOURCE DEFINITIONS
* INITIAL CONDITIONS
* RUN CONTROLS
* END

The END card must be the last card appearing. If it is not present, the end of file card is interpreted as the END card.

The run controls card group must be present. The source definition card group must be present unless the impulse or step response was specified in the run controls card group. If not execution will be terminated.

Subroutine: SRCINT

Description:

This subroutine interprets the source definition card group for the matrix mode. This card group is signaled by a card with an asterisk in column 1, followed by the words "SOURCE DEFINITIONS" in free format. The subroutine assumes all cards following this card up to the next card with an asterisk in column 1 are source definition cards. These cards are of the form

NAME = $f_1(t) + f_2(t) + \ldots + f_N(t)$

where NAME is the name of a source defined in the circuit description, and each $fi(t)$ is one of the following

```
A
A*IMP
A*SIN(B*T)
A*SIN(B*T+C)
A*COS(B*T)
A*COS(B*T+C)
```

where A, B, and C are floating point numbers in free format. The asterisks indicating multiplication need not appear. A source description may be continued on additional cards by placing a dollar sign ($) in column 1 of the cards on which the description is continued as long as each $f_i(t)$ is completely described on only 1 card and the total number of functions describing a given source is no more than 20. A sinusoidal or cosinusoidal function with a phase angle is broken into a sinusoidal plus a cosinusoidal function with zero phase angle thus counting as two functions.

This subroutine will terminate program execution if an error occurs in a source description or if not all the sources defined in the circuit description are described in the source definition cards.

Additional Input Parameters

NIN - The number of circuit inputs.

NAMIN - A vector containing the names of the circuit inputs in the order they appear in the state variable equations.

Output Parameters

NORDER - A vector relating the order the sources are defined in the state variable equations with the order they are defined in NSORCE, ISORCE, and SOURCE.

ISORCE - The number of functions describing source I is ISORCE(K) where K = NORDER(I)

NSORCE - The function type for the Jth function describing the Ith source is prescribed by NSORCE(K,J) where K = NORDER(I), and

NSORCE(K,J) = 1 step

NSORCE(K,J) = 2 sine

NSORCE(K,J) = 3 cosine

NSORCE(K,J) = 4 impulse

SOURCE - The magnitude of the Jth function of the Ith source is contained in SOURCE(K,J,1), where K = NORDER(I) and the angular velocity (for sine and cosine functions) is SOURCE(K,J,2).

Machine Dependance - This subroutine is machine independent but calls the machine dependent subroutines APACK and ATODEC.

Subroutine: RUNCRL

This subroutine reads in the run controls card group for the matrix method.

The completion of this card group is signaled by a card with an asterisk in column 1 or the end of file card.

Input Parameters

The run controls cards read must be one of the following

INITIAL TIME = A

STEP RESPONSE

IMPULSE RESPONSE

PRINT START = A

STOP TIME = A

POINTS PRINTED = N

PLOT INCREMENT = A

where A is a floating point number and N is an integer.

Output Parameters

TO	Initial value of time
TSTART	Time at which output printings is to begin.
NPTS	Number of output points to be printed.
NPUT	Type of input
DT	Time between output printings.

Default Options

If no initial time is specified, it is assumed to be zero.

If no print start time is specified, it is assumed to be the initial time.

If a stop time but no plot increment or points printed is specified or if only the plot increment is specified, 100 time points will be printed.

Machine Dependance

This subroutine is machine independent but calls the machine dependent routines APACK and ATODEC.

Program: CROLCM

This program calls COMPAK to rearrange the variables in blank common from the form supplied by CORNAP to the form required by GEAR and MATRIX. One of the subroutines CROLGR or CROLMX is then called to read the card groups peculiar to the GEAR or the MATRIX mode.

Program INTGR

This subroutine performs the integration of the state variable equations.

The upper limit of the integration may be given by specifying the maximum value of the independent variable time or by specifying the number of points printed.

If the maximum value of time is the stop condition, interpolation is used to determine the values of the state variables at the stop time.

The following error conditions will terminate program execution

1. Requested error is smaller than can be handled for this problem.
2. Corrector convergence could not be achieved with H greater than HMIN.
3. The requested error could not be achieved with H equal to HMIN.

Input Parameters

NSTV	The number of state variable equations.
TO	The lower integration limit.
XO	A vector containing the values of the state variables measured at time TO.
TSTART	The time at which printings is begun.
TSTOP	The upper integration limit
MAXPTS	The maximum number of time points to be printed.
ISTOP	The stop condition

ISTOP = 1 Maximum time points printed.

ISTOP = 2 Maximum value of time.

NO The number of circuit outputs

NAMOUT A vector containing the names of the circuit outputs

H The initial integration step size.

HMIN The minimum integration step size.

HMAX The maximum integration step size permitted.

INCRMT The number of integration steps between printings.

EPS The euclidean norm of a vector whose Ith element is the single step error of the Ith state variable divided by the maximum value of the Ith state variable previously encountered must be less than this value.

METHOD The integration method used.

METHOD = 0 An Adams predictor corrector method.

METHOD = 1 A multistep method suitable for stiff systems.

MAXDER The maximum derivative that should be used in the method. It must be less than 8 for Adams and 7 for stiff methods.

Subroutine: DATPNT

This subroutine prints the output for the Gear mode. It performs the calculation

$$\underline{Y}(t) = \underset{\sim}{C}\,\underline{x}(t) + \underset{\sim}{D}\,\underline{u}(t)$$

where $\underline{Y}(t)$ is the output vector at time T. $\underline{x}(t)$ is a vector containing the values of the state variables at time T, and $\underline{u}(t)$ is a vector containing the values of the inputs at time T.

Input Parameters:

T - present value of time.

x - a 8 x 30 matrix. x(1,I) contains the value of the Ith state variable measured at time T.

c,D - constant matrices of the output equation.

NO - number of circuit outputs.

NSTV - number of state variable equations.

NIN - number of inputs defined in the circuit description.

NORDER- a vector telling the order the inputs are defined in USEFCN.

Output Parameters:

This subroutine prints the value of time and the values of the outputs measured at this time.

Subroutine Required: USEFCN

Machine Dependency: None

Subroutine: DIFFUN(T,X,DX)

This subroutine evaluates the derivatives of the dependent variables X, with respect to the independent variable T, using the matrix equation

$$\dot{X}=AX+BU$$

Input Parameters

NSTV	The number of state variable equations.
NIN	The number of circuit inputs
A	System matrix
B	Input matrix

T   Present value of time

$\underline{X}$ X(1,I) contains the value of the Ith state variable measured at time T.

U   U(I) is the value of the NORDER(I)TH input measured at time T

NORDER   NORDER(I) is where the Kth source is located

Output Parameters

DY   DY is a vector containing $\dot{X}$ measured at the present values of X and T.

Subroutines Required

USEFCN

Soubroutine: PEDERV

Description: This subroutine supplies DIFSUB with the Jacobian matrix of the state variable equation:

$$\underline{\dot{X}} = \underset{\sim}{A}\,\underline{x} + \underset{\sim}{B}\,\underline{u}$$

which is, in fact, the A matrix.

Input Parameters:

M - number of state variable equations.

A - system matrix stored in core with the first column of A in the 1st 30 storage locations, the second column of A in the next 30 storage locations and so forth.

Output Parameters:

PW - Jacobian matrix. This is the system matrix, A, with each column stored in core starting in the core location following the end of the previous column.

<u>Subroutines Required:</u> None

<u>Machine Dependancy:</u> None

<u>Example of Usage:</u>

Given the system matrix

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

A is supplied as

$$1,2,3,\overbrace{b,b,\ldots,b}^{27\text{ blanks}},4,5,6,\overbrace{b,b,\ldots,b}^{27\text{ blanks}},7,8,9,\overbrace{b,b,\ldots,b}^{810\text{ blanks}}$$

PW is outputted as

$$1,2,3,4,5,6,7,8,9,\overbrace{b,b,\ldots,b}^{891\text{ blanks}}$$

<u>Subroutine:</u> MATINV(A,N,JSING)

Purpose

Invert a matrix

Description of parameters

A - Input matrix, destroyed in computation and replaced by resultant inverse.

N - Order of matrix A

JSING - Indicates whether A is singular; one if not, minus one if it is.

Remarks

Matrix A must be a general matrix

Subroutines and function subprograms required: None

Method

The standard Gauss-Jordan method is used. The determinant is also calculated. A determinant of zero indicates that the matrix is singular.

Subroutine: DECOMP(N,PSAVE,J1)

Function:

This routine performs an LU decomposition of the NxN matrix found in PSAVE. PSAVE is replaced by the matrices U and L, with U the main diagonal and above, L below the main diagonal. All main diagonal elements of L are unity, so need not be stored.

If the PSAVE matrix is nonsingular, the routine returns a value of +1 for J1; if singular, J1 is set to -1.

The routines DECOMP and SOLVE are lifted directly from

Forsythe and Moeller, Numerical Linear Algebra, p. 68

and are well discussed in this reference.

Subroutine: SOLVE(N,PSAVE,SAVE,N5)

Function.

This routine employs the LU decomposition of DECOMP stored in PSAVE to solve for $\overline{X}$ the N equations

$$(LU)\overline{x} = \overline{b},$$

with various $\overline{b}$ supplied by Gear's subroutine. The result is stored in SAVE. N5 is a parameter used for addressing the proper locations in SAVE in which $\overline{x}$ is to be stored

For large systems (N greater than 5) the LU decomposition is faster than the (MATINV) matrix inversion previously used by Gear If desired, minor alterations to STICAP will restore the Gear routine to its previous form. These modifications are stated in Gear's ACM DIFSUB description (see STICAP theory manual for the complete reference).

Program: MATRIX

This overlay solves the system of state variable equations by performing a spectral decomposition of the transition matrix then integrating in closed form. This technique is described in the paper:

"An Efficient Matrix Algorithm for the Solution of Linear Systems with Widely Separated Eigenmodes" ... by E. Young, J. Heinbockel, and M. Ransom, Proceedings of the 4th Asilomar Conference on Circuit and System Theory, Pacific Grove, Calif., 1970.

Inputs to this program are described in the comments of the three secondary overlays and in the programmer's manual.

This first secondary overlay determines the eigenvalues of the system matrix, determines the inverse of the Vandermonde matrix given by these eigenvalues, and finally determines a sparse matrix code for the system matrix.

```
CALL OVERLAY(3HCOM,4,1,0)
```

The next secondary overlay forms the transition matrix times a vector

```
EXP(A*T)*V
```

as a finite series for the initial condition vector and for each driving vector. This expression is then integrated in close form and the symbolic solution evaluated at each time point at which output is requested.

```
CALL OVERLAY(3HCOM,4,2,0)
```

This final secondary overlay prints the tabulated results, and if requested, the close form solution.

```
CALL OVERLAY(3HCOM,4,3,0)
```

<u>Program:</u> SETSVE

This program calls EIGNV to determine the eigenvalues of the system matrix and VANINV to form the inverse of the Vandermonde matri given by these eigenvalues. Finally this program determines a sparse matrix code for the system matrix.

<u>Input Parameters:</u>

A - The system matrix.

N - The number of state equations.

EPS - The value below which an element is felt to be zero.

<u>Output Parameters:</u>

C - A complex matrix containing the inverse of the Jacobian matrix.

KNZ - The number of non-zero elements in the system matrix.

NZR,NZC - Vectors containing a sparse matrix code for the syster matrix. The Ith non-zero element of the system matri is located in row NZR(I) and column NZC(I).

ANZ - A vector which replaces A in core which contains only the non-zero elements of A.

<u>Subroutine:</u> VANINV

This subroutine forms the inverse of the Vandermonde matrix given by the eigenvalues of the system matrix. The Vandermonde matrix of the n elements $\lambda_1, \lambda_2, \ldots, \lambda_n$ is the n x n matrix

$$\underset{\sim}{V} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \ldots & \lambda_n^2 \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \ldots & \lambda_n^{n-1} \end{bmatrix}$$

The inverse of the Vandermonde matrix is formed in this subroutine using a technique due to Kaufman.[1]

Let $\quad [C_{ij}] = [V_{ij}]^{-1}$

then
$$C_{ij} = \frac{\sum_{k=0}^{n-j} a_k \lambda_i^{\,n-j-k}}{\prod_{\substack{k=1 \\ k \neq i}}^{n} (\lambda_i - \lambda_k)}$$

where the a_k are the coefficients of the characteristic equation

$$P(\lambda) = \prod_{i=1}^{n} (\lambda - \lambda_i)$$

$$= a_0\lambda^n + a_1\lambda^{n-1} + \ldots + a_{n-1}\lambda + a_n$$

In this subroutine the a_k are formed from the recursive relationship

$$a_0 = 1$$

$$a_k = -\frac{1}{k} \sum_{m=1}^{k} a_{k-m} T_m$$

where
$$T_m = \sum_{P=1}^{n} \lambda_P^m$$

We note finally that if $\lambda_i = \lambda_j^*$, then $C_{ik} = C_{jk}^*$.

<u>Input Parameters</u>:

N - The number of state variable equations.

ROOT - A complex vector whose elements are the eigenvalues of the system matrix.

<u>Output Parameters</u>:

C - A complex matrix containing the Vandermonde inverse.

---

1. Kaufman, I., "Evaluation of an Analytical Function of a Companion Matrix with Distinct Eigenvalues," 1969 Proceedings of IEEE Letters, pp. 1180-1181.

Subroutine: HESEN(A,M)

Supplier: NASA Routine*

This subroutine reduces a matrix through similarity transformations to an upper-Hessenburg matrix. An upper-Hessenburg matrix is one in which all elements below the subdiagonal are zero.

This subroutine is the NASA Langley Research Center routine HESSEN.

Input Parameters:

M - The order of the matrix being reduced.

A - The matrix being reduced. This matrix is destroyed in the computations.

Output Parameters:

A - The reduced matrix which has replaced in input matrix in core.

Subroutine: QRTRAN(A,N,R,SIG,D)

Supplier: NASA Routine

This subroutine performs one double Q-R transformation on the matrix A.

Let

$$A = Q_1R_1$$

$$A_1 = R_1Q_1 = Q_1^{-1} AQ_1 = Q_2R_2$$

$$A_2 = R_2Q_2 = Q_2^{-1} A_1Q_2 = Q_2^{-1} Q_1^{-1} AQ_1Q_2$$

where

Q_1 and Q_2 are unitary matrices

and

R_1 and R_2 are right triangular matrices.

---

*Not contractor supplied.

A is then replaced in core by A_2.

This subroutine is the NASA Langley Research Center routine QR2.

<u>Subroutine</u>: EIGNV

<u>Supplier</u>: NASA Routine

This subroutine finds the eigenvalues of a real, non-symmetric matrix using the double Q-R transformation.

This routine was slightly modified from the NASA Langley Researc Center routine REIS.

<u>Input Parameters</u>:

KJN - The number of state variable equations.

AO - The matrix whose eigenvalues are being determined.

<u>Output Parameters</u>:

ROOT - A complex vector containing the eigenvalues of the input matrix with real eigenvalues stored first and each complex eigenvalue followed immediately by its complex conjugate.

<u>Program</u>: INTCL

This overlay calls the various integration routines to perform the closed form integration for each function making up each circuit input. The close form solution is then evaluated at each time point requested.

<u>Input Parameters</u>:

N	The number of state variables.
NPTS	The number of time points to be printed.

TSTART	The time at which output printings are to be begun.
DT	The time increment between output printings.
A,B,C,D	Matrices in the input and output equations.
NIN	The number of circuit inputs.
NO	The number of circuit outputs.
XO	A vector containing the initial conditions of the state variables.
INZ	Tells whether all initial conditions are zero. INZ = 1 if there are any non-zero initial conditions.
NZR,NZC,KNZ	Variables containing a sparse matrix code for the system matrix.
IPRINT	Tells whether or not the closed form solution i to be printed.
NORDER	A vector relating the way the sources are defined in ISORCE, NSORCE and SOURCE with the way they are defined in the output and state variable equations.
ISORCE,SOURCE, NSORCE	Matrices containing information about sources. See SCEVAL for description.
NSC	The number of different angular velocities exciting the circuit.

<u>Output Parameters</u>:

OUT	A matrix containing the outputs evaluated at each time point.

Subroutines required in this overlay:

STEADY, INIVAL, STORE, TRANS, SCEVAL

EAT, SVESTP, SVESIN, SVECOS, MPROD

Subroutine: SVECOS(AMPL,OMEGA)

This subroutine determines the response of the system due to cosine driving functions.

Input Parameters:

N The number of state variable equations.

NR The number of real eigenvalues of the system matrix.

AMPL The amplitude of the driving function.

OMEGA The angular velocity of the cosine driving function.

P A NXN complex matrix whose columns are the V'S of the expression

```
EXP(A*T)*R = V1*EXP(LAMBDA(1)*T)
           + V2*EXP(LAMBDA(2)*T)
           + ...
           + VN*EXP(LAMBDA(N)*T)
```

where

A is the system matrix.

R is the driving vector.

LAMBDA(I) is the Ith eigenvalue of A.

and

T is time.

TO The time at which the initial conditions of the system were measured.

ROOT A complex vector of length N containing the eigenvalues of the system matrix.

Output Parameters:

CFCOS A NXN matrix containing the transient part of the solution.

STYCOS A NX2 rectangular matrix containing the steady-state portion of the solution.

Subroutine: SVESIN(AMPL,OMEGA)

This subroutine performs the integration of EXP(A(T-TAU))*R *SIN(OMEGA*X) from TO to t where A is a matrix and R is a vector.

Input Parameters:

N The number of state variables.

OMEGA The angular velocity of the driving function.

R A vector of N length associated with the driving function.

TO The initial value of time.

ROOT A vector containing the circuit eigenvalues.

P A matrix containing EXP(A*T)*R.

NR The number of real eigenvalues.

AMPL The amplitude of the driving function.

Output Parameters:

STYSIN A NX2 rectangular matrix containing the steady state solution to the integration.

COEF A NXN matrix containing the transient part of the solution.

Subroutine: SVESTP(AMPL)

This subroutine determines the responses due to step driving sources.

Input Parameters:

N . The number of state variable equations.

NR The number of real eigenvalues of the system matrix.

ROOT A complex vector of length N containing the eigenvalues of the system matrix.

TO The time at which the initial values of the state variables are measured.

P A NXN complex matrix whose columns are the V'S of the expression

EXP(A*T)*R = V1*EXP(ROOT(1)*T)
+ ...
+ VN*EXP(ROOT(N)*T)

where

A is the system matrix

R is the driving vector

ROOT(I) is the Ith eigenvalue of A

and

T is time.

Where complex conjugates occur, the complex conjugate V is omitted and the remaining V'S stored consecutively in P

AMPL The amplitude of the step function.

Output Parameters:

COEF A NXN matrix containing coefficients corresponding to the transient part of the response.

STP A vector of length N containing the steady state dc part of the response.

Subroutine: INIVAL(AMPL)

This subroutine is called to determine the solution correspondi[ng] to the initial conditions or for impulse driving functions.

Input Parameters:

N Number of state variables.

ROOT Vector containing the eigenvalues of the system matrix.

NR Number of real eigenvalues.

P A matrix containing EXP(A(T-TO)) where A is the system matrix.

AMPL The amplitude of the impulse function.

Output Parameters:

COEFO A matrix containing the closed form solution.

Subroutine: MPROD(A,B,L,M,N)

This subroutine multiplies a matrix A into a matrix B and stores the product in B.

Input Parameters:

A A matrix dimensioned 10x30.

B A matrix dimensioned to have 30 rows but having a variable number of columns. This matrix is destroyed during the computation.

L Number of rows in matrix A.

M Number of columns in matrix A and rows in matrix B.

N Number of columns in matrix B.

Output Parameters:

B A matrix containing the product.

Subroutine: EAT(V)

This subroutine finds the expression

$$e^{\underset{\sim}{A}t} \cdot \underline{V}$$

where $\underset{\sim}{A}$ is the system matrix and $\underline{V}$ is a given vector. This expression is given as

$$e^{\underset{\sim}{A}t} \cdot \underline{V} = \underline{V}_1 e^{\lambda_1 t} + \ldots + \underline{V}_n e^{\lambda_n t}$$

Here

$$\underline{V}_i = \sum_{k=1}^{n} C_{ik} \underset{\sim}{A}^{k-1} \underline{V},$$

$\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $\underset{\sim}{A}$ and $[C_{ij}]$ is the inverse of the Vandermonde matrix given by these eigenvalues. Since if $\lambda_j = \lambda_i^*$, $\underline{V}_j = \underline{V}_i^*$ complex conjugate $\underline{V}$'s are not given.

Input Parameters:

N	The number of state variable equations.
NR	The number of real eigenvalues in the system matrix.
A	A vector containing the non-zero elements of the system matrix.
KNZ	The number of non-zero elements in the system matrix.
NZR,NZC	Two vectors containing a sparse matrix code for the system matrix. The Ith non-zero element in the system matrix is in row NZR(I) and column NZC(I).
C	A complex matrix containing the inverse of the Vandermonde matrix given by the eigenvalues of the system matrix.

Output Parameters:

P	A matrix whose columns are the $\underline{V}$'s in the above expansion except that if $\underline{V}_i$ is complex, Re$[\underline{V}_i]$ is stored

in the Ith column of $\underset{\sim}{P}$ and $Im[\underline{V}_1]$ is stored in the I+1th column of $\underset{\sim}{P}$. $\underline{V}_{1+1}$ which is $\underline{V}_1$* is then omitted.

Subroutine: SCEVAL

This subroutine evaluates the expression

D*U

where D*U is the matrix expression in the output equation

Y = C*X + D*U

at each time point at which output is requested and adds the result to the contents of a matrix out which contains C*X at each output time point.

Input Parameters:

TSTART The time at which output printings are to start.

DT The time increment between output printings.

NUM The number of output time points to be printed.

NIN The number of circuit inputs.

NO The number of circuit outputs.

NORDER A vector relating the way the sources are defined in ISORCE, NSORCE, and SOURCE with the way they are ordere in the state and output equations.

ISORCE A vector whose Ith element tells the number of functior making up the NORDER(I) source.

NSORCE A matrix whose I,J element tells the type of the Jth function making up the NORDER(I) source.

NSORCE(I,J)=1 STEP

NSORCE(I,J)=2 SINE

NSORCE(I,J)=3 COSINE

NSORCE(I,J)=4 IMPULSE

SOURCE A 3-dimensional matrix whose I,J,1 element tells the magnitude of source NSORCE(I,J), and whose I,J,2 element tells (for sine and cosine functions) the angular velocity of the function.

D The matrix relating the outputs to the inputs in the output equation.

Output Parameters:

OUT A matrix containing the circuit outputs at the requested time points.

Subroutine: TRANS

This subroutine calculates the transient portion of the solution by substituting into the closed form expressions for the solutions.

Input Parameters:

N The number of state variable equations.

NUM The number of different time values at which the output is to be measured.

TSTART The time at which the first measurement is to be made.

COEF A matrix containing the coefficients of the transient portion of the solution.

ROOT A complex vector of N length containing the eigenvalues of the system matrix.

NR The number of real eigenvalues of the system matrix.

DT The interval between time samples.

Output Parameters:

OUT A matrix containing the outputs at the requested time points.

The values of the outputs are given at time

TSTART+K*DT, K=0,1,...,NUM-1

Additional Subroutines Required: None

Subroutine: STEADY(NTYPE,OMEGA)

This subroutine calculates the steady state portion of the solution by substituting into the closed form solution. This calculation is made at time

TO + K*DT   K = 0,1,...,NUM-1

where TO is the time at which printing is to begin,
NUM is the number of time points at which printing is to be made,
and DT is the printing increment.

Input Parameters:

- N — The number of state equations.
- NUM — The number of different time points at which printing is to be made.
- TSTART — The time at which the first printing is to be made.
- NTYPE — The type of driving function.
  - NTYPE = 1 Sine or cosine driving function.
  - NTYPE = 2 Step function.
- STYSC — A matrix containing the closed form steady state portio of the solution for sine and cosine driving functions..
- STP — A vector containing the closed form steady state portio of the solution for step functions.

Output Parameters:

- OUT — A matrix containing the circuit outputs at the requeste time points.

Subroutine: STORE(OMEGA)

This subroutine stores the steady-state response for sine and cosine driving functions. The subroutine will combine responses with the same angular velocities.

Input Parameters:

NO The number of circuit outputs.

STYSC A matrix whose I,1 element contains the steady-state sine response for the Ith output and whose I,2 element contains the corresponding cosine response.

OMEGA The angular velocity of the driving function.

Output Parameters:

NSC The number of different angular velocities encountered so far.

ANGVEL A vector whose Ith component contains the angular velocity of the Ith sine or cosine steady-state response

SICO A matrix whose 1,J,K element contains the steady-state response of the Kth output with the Jth angular velocity and whose 2,J,K element contains the corresponding cosine response.

Program: OUTPNT

This overlay is used for printing output information for the matrix mode. Tabulated results are always printed. Close-form solutions are printed only if requested.

Input Parameters:

IPRINT Tells whether or not the closed form solution is to be printed.

See CLSOL and TABPNT for additional input parameters.

Subroutine: CLSOL

This subroutine prints the closed form solution for the requested outputs.

Input Parameters:

NO The number of circuit outputs.

NAMOUT A vector containing the names of the circuit outputs.

NSC The number of sine and cosine inputs.

ANGVEL A vector whose Ith element is the angular velocity of the Ith sine or cosine input.

SICO A matrix whose 1,J,I element tells the steady-state sinusoidal response of the Ith output due to the Jth sine or cosine input, and whose 2,J,I element tells th corresponding steady state cosinusoidal response.

STP A vector whose Ith element gives the steady-state ster response for the Ith output.

N The number of state variables.

NR The number of real eigenvalues.

ROOT A complex vector containing the eigenvalues of the sys matrix with the real eigenvalues stored first and eact complex eigenvalue followed immediately by its comple: conjugate.

COEF A matrix containing the transient solution. The transient solution for the Ith output is·

COEF(I,J)*EXP[ROOT(J)*T]

for ROOT(J) a real eigenvalue. When ROOT(J) is a com eigenvalue the solution corresponding to ROOT(J) is combined with the solution corresponding to the complex conjugate of ROOT(J) to give:

EXP[RE(ROOT(J))*T]

* [COEF(I,J)*SIN(IM(ROOT(J))*T)

+COEF(I,J+1)*COS(IM(ROOT(J))*T)]

where RE(ROOT(J)) is the real part of ROOT(J) and IM(ROOT(J)) is the imaginary part of ROOT(J).

Subroutine: TABPNT

This subroutine prints the tabulated results for the matrix mode.

Input Parameters:

TSTART	Time at which printing is to begin.
DT	Time increment between output printings.
NPTS	The number of time points to be printed.
OUT	A matrix whose I,J element is the Jth output measured at the Ith time value.
NO	The number of circuit outputs.
NAMOUT	A vector containing the names of the circuit outputs.

Subroutine: ATAN3(X,Y)

This subroutine combines a sine function with a cosine function where both functions have equal angular velocities but no phase angle into a sine function with a phase angle. The phase angle is chosen to be positive or negative such as to make the magnitude of the phase angle less than pi/2.

<u>Input Parameters</u>:

X The coefficient of the sine function.

Y The coefficient of the cosine function.

<u>Output Parameters</u>:

X The coefficient of the resulting sine function.

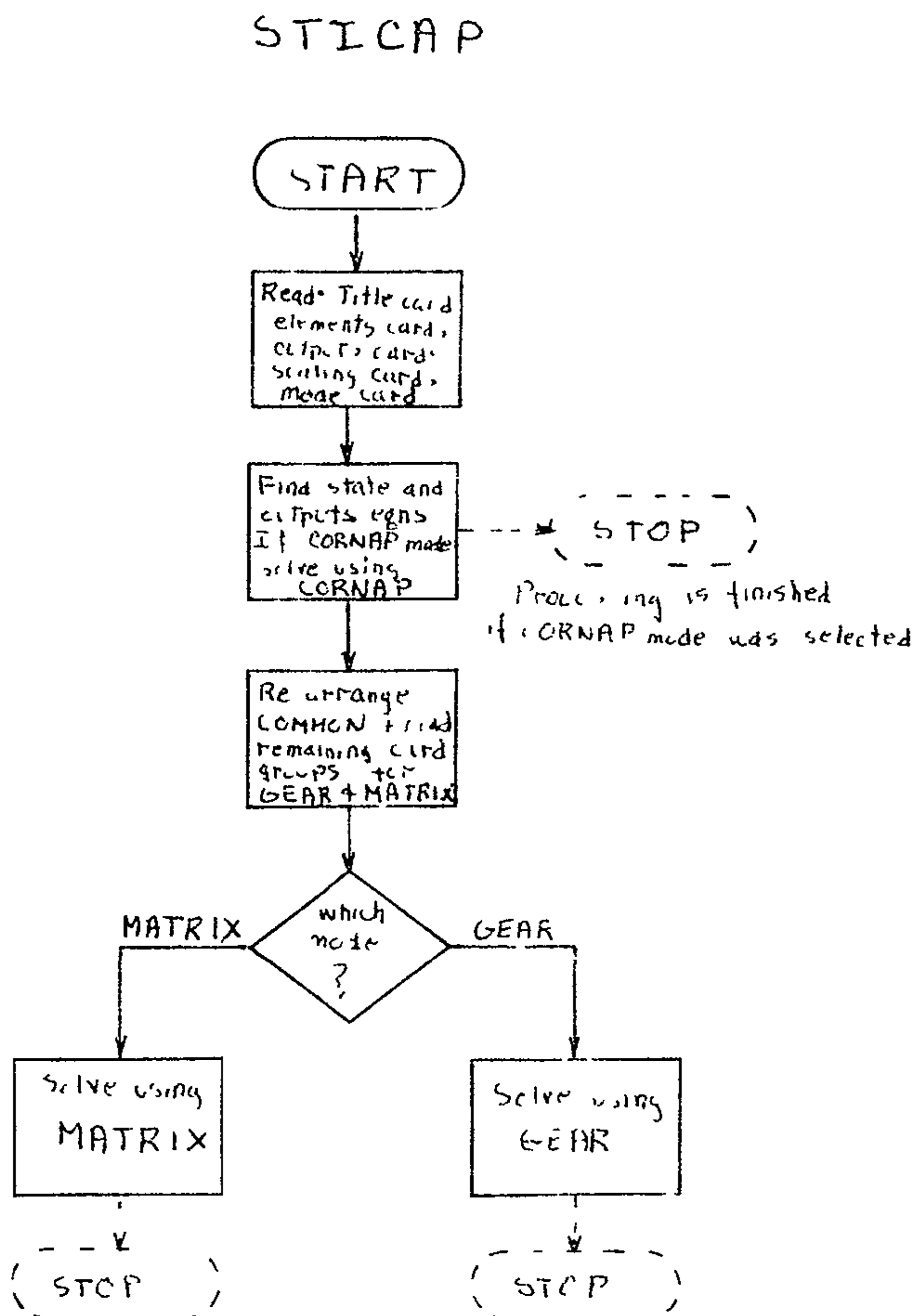
Y The phase angle in radians of the resulting sine function.

Note: Both input parameters are destroyed in the computations.

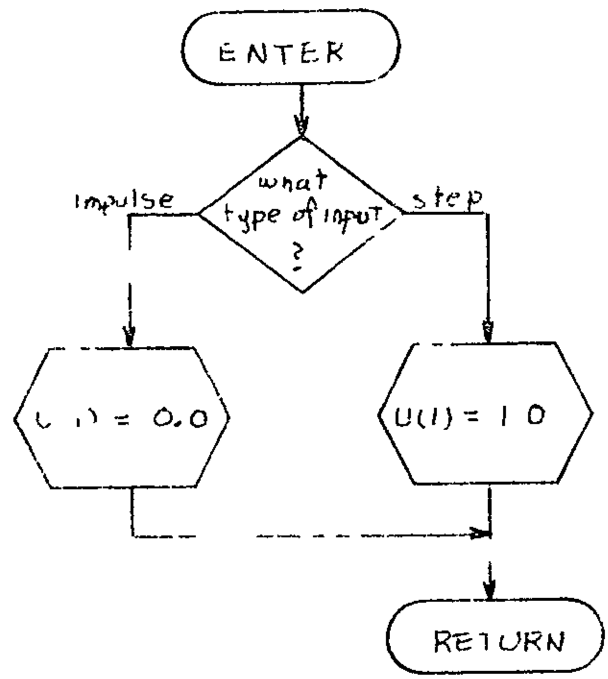
CHAPTER VI

FLOWCHARTS

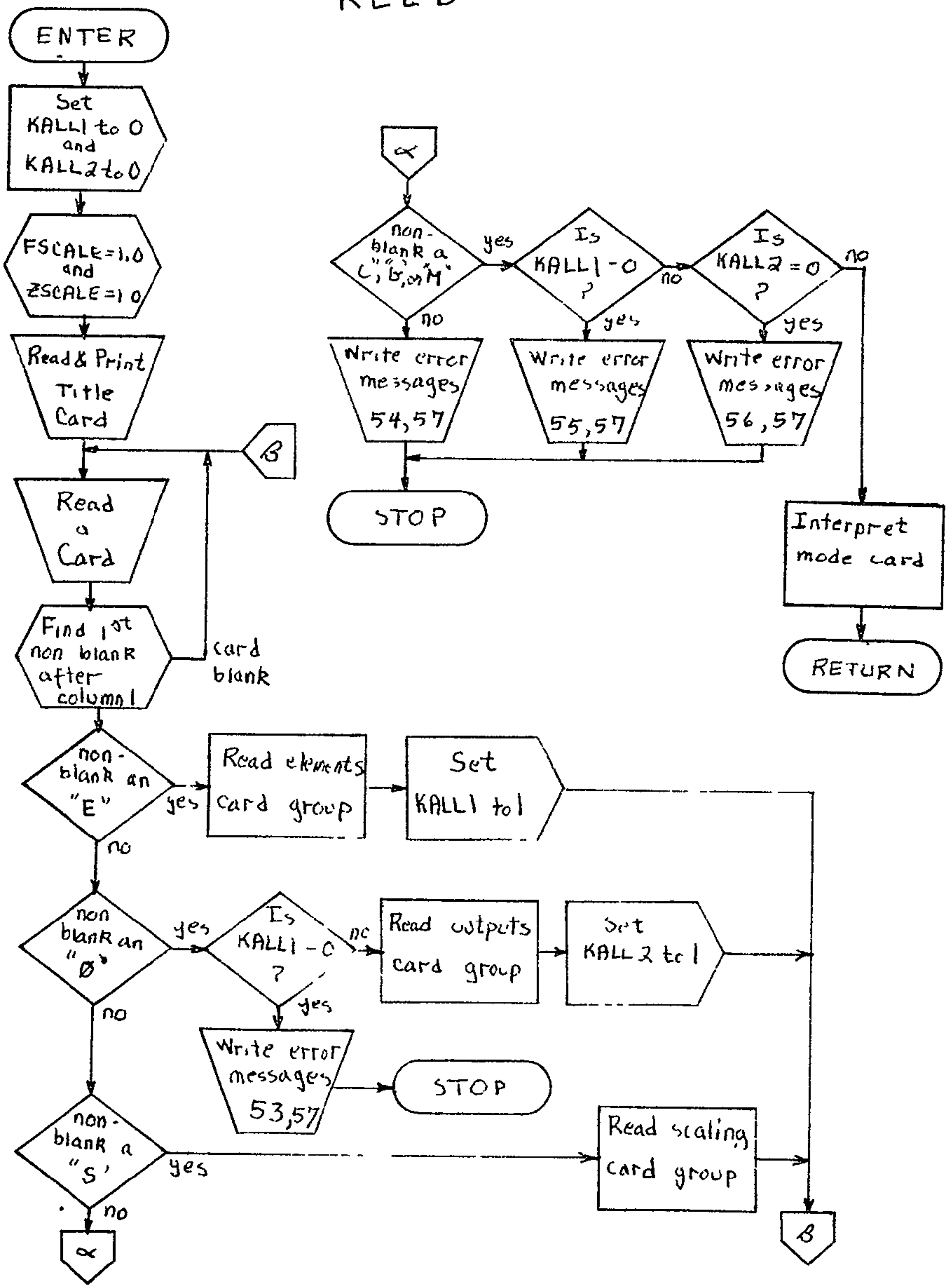
This chapter contains the flowcharts for all contractor supplied subroutines.


STICAP
START
Read: Title card elements card, outputs card, scaling card, mode card
Find state and outputs eqns If CORNAP mode solve using CORNAP
STOP
Processing is finished if CORNAP mode was selected
Re arrange COMMON + read remaining card groups for GEAR + MATRIX
which mode ?
MATRIX
GEAR
Solve using MATRIX
Solve using GEAR
STOP
STOP

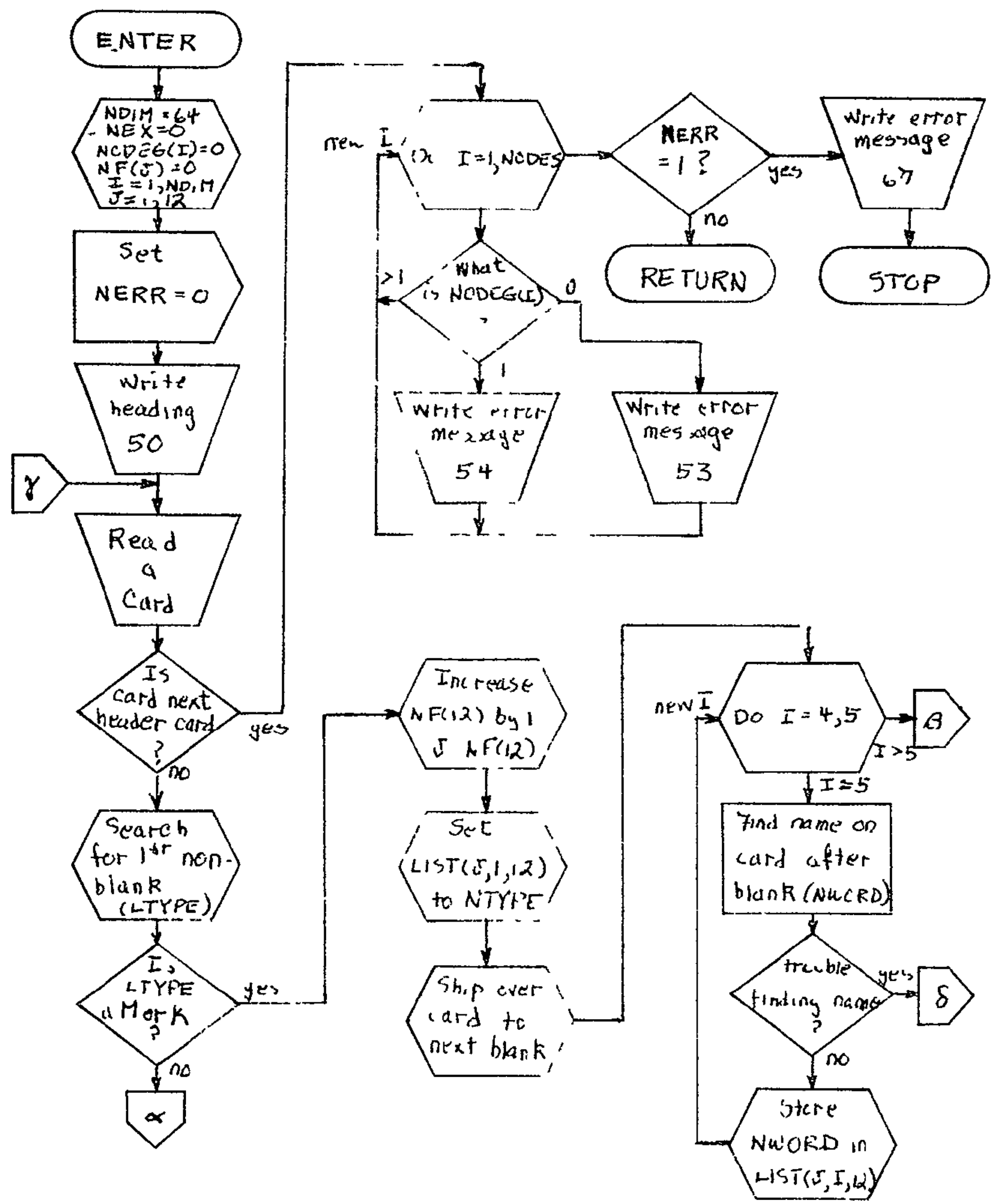
USEFCN



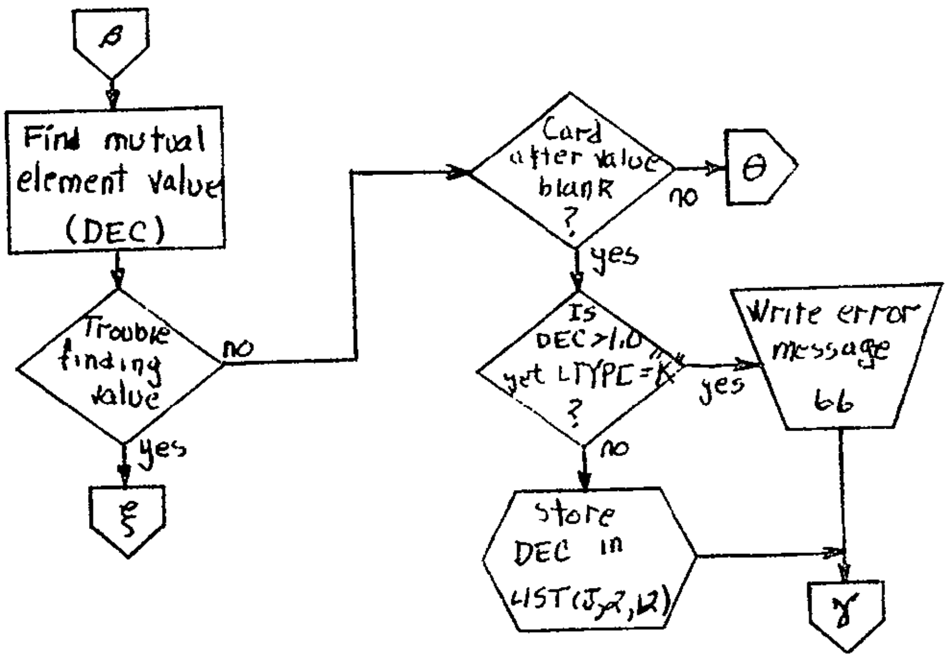
REED

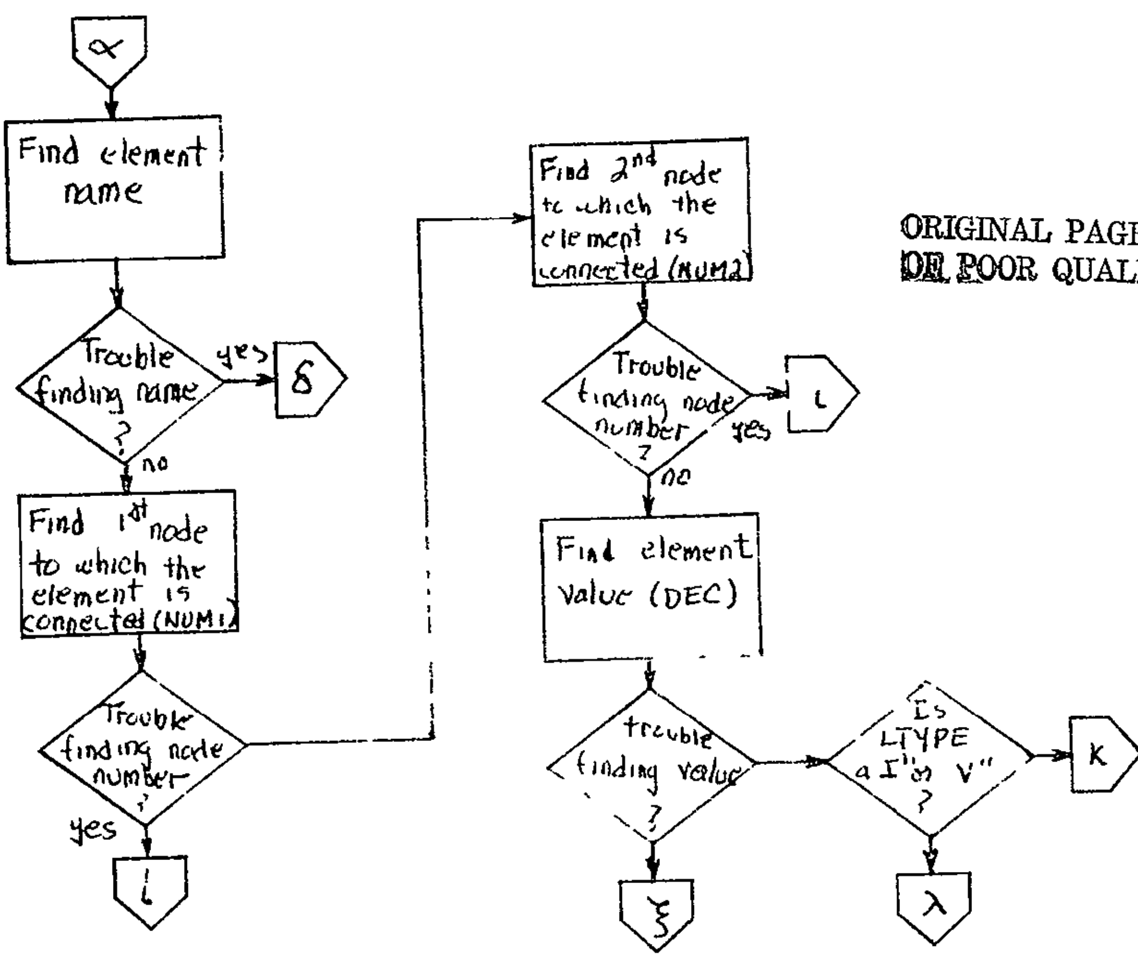

ENTER
Set KALL1 to 0 and KALL2 to 0
FSCALE=1.0 and ZSCALE=1.0
Read & Print Title Card
β
Read a Card
Find 1st non blank after column 1
card blank
non-blank an "E"
yes
Read elements card group
Set KALL1 to 1
no
non blank an "Ø"
yes
Is KALL1 - 0 ?
no
Read outputs card group
Set KALL2 to 1
yes
Write error messages 53,57
STOP
no
non-blank a "S"
yes
Read scaling card group
no
α
β
α
non-blank a "C", "B", or "M"
yes
Is KALL1 - 0 ?
no
Is KALL2 = 0 ?
no
no
yes
yes
Write error messages 54,57
Write error messages 55,57
Write error messages 56,57
STOP
Interpret mode card
RETURN

INPUT

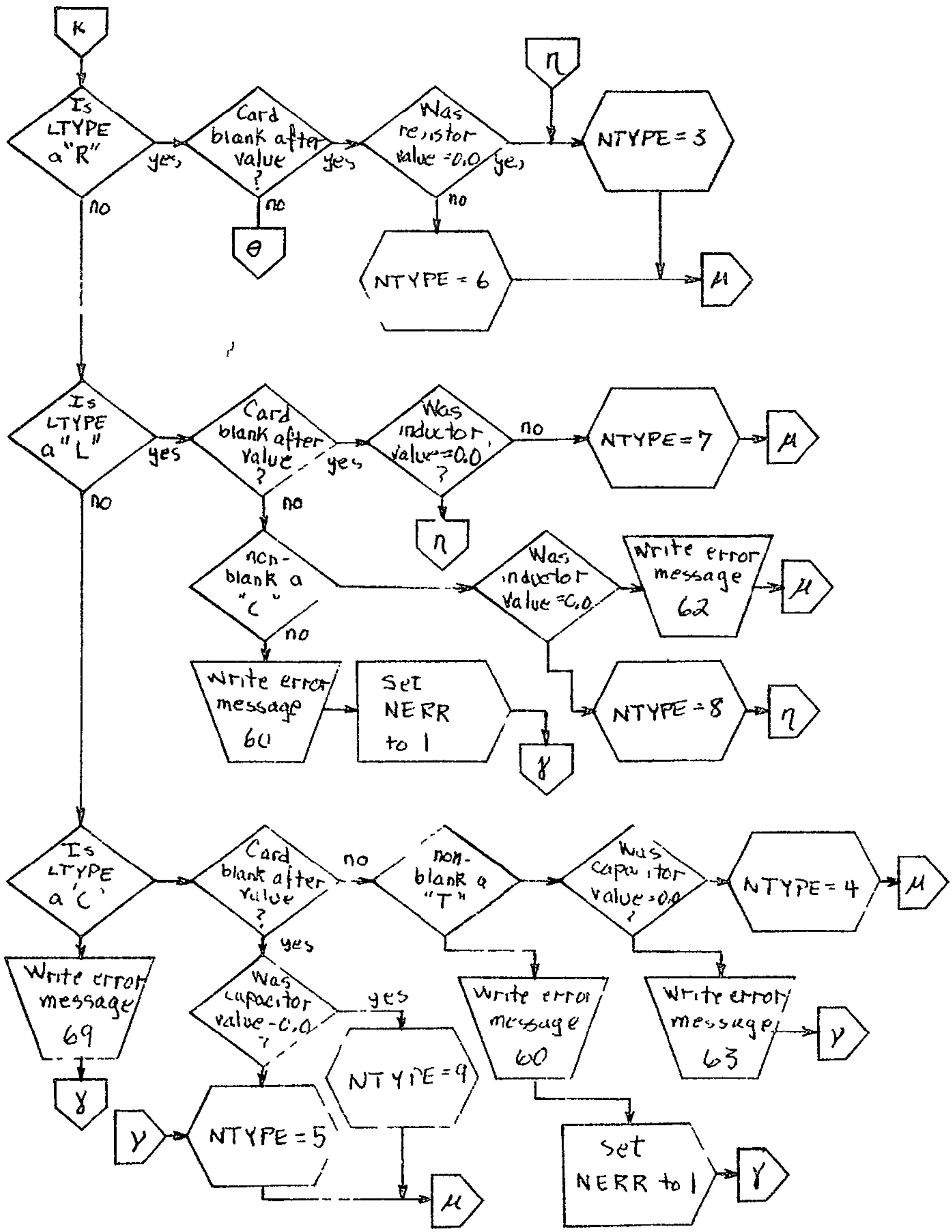


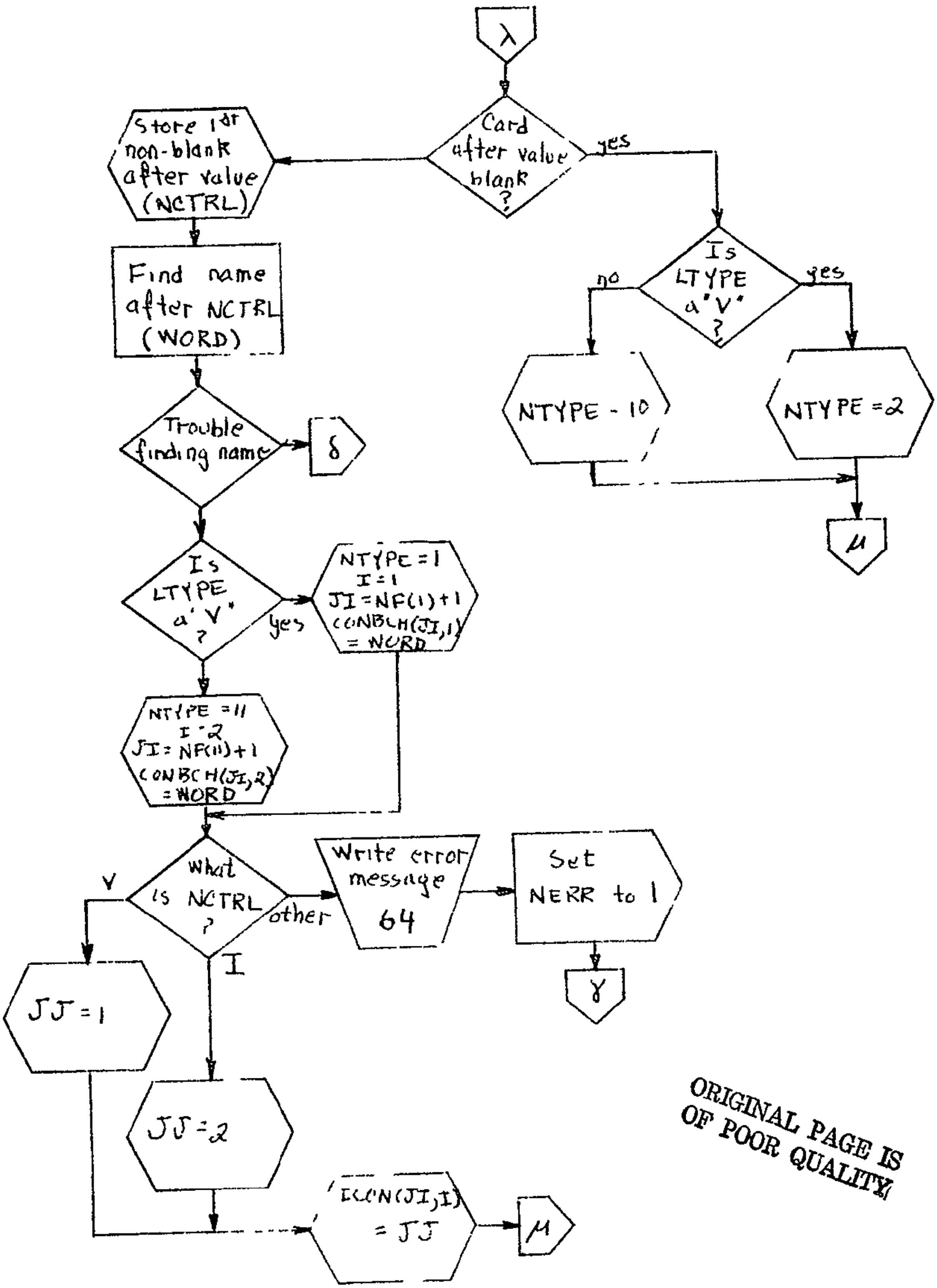
ORIGINAL PAGE IS
OF POOR QUALITY

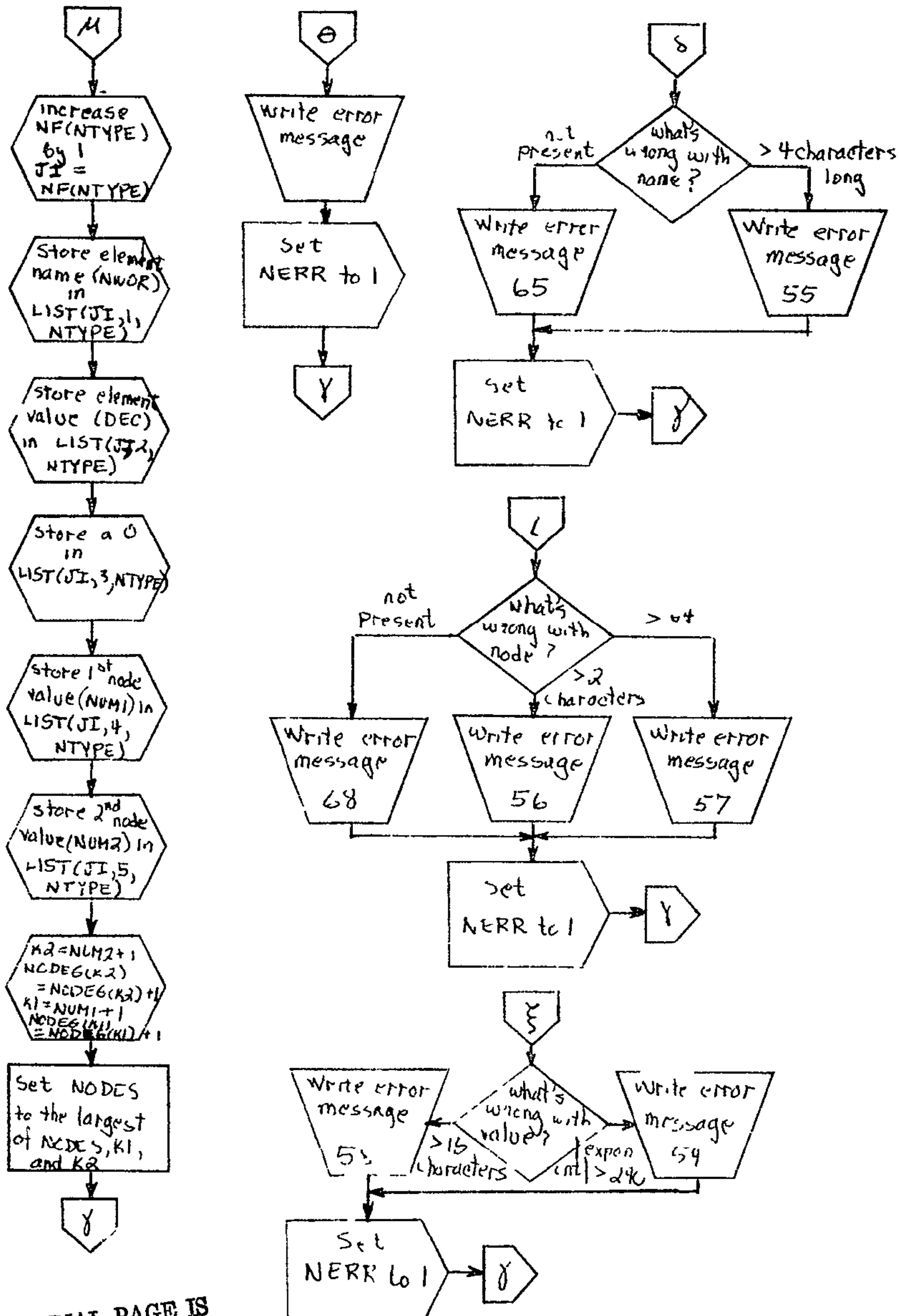




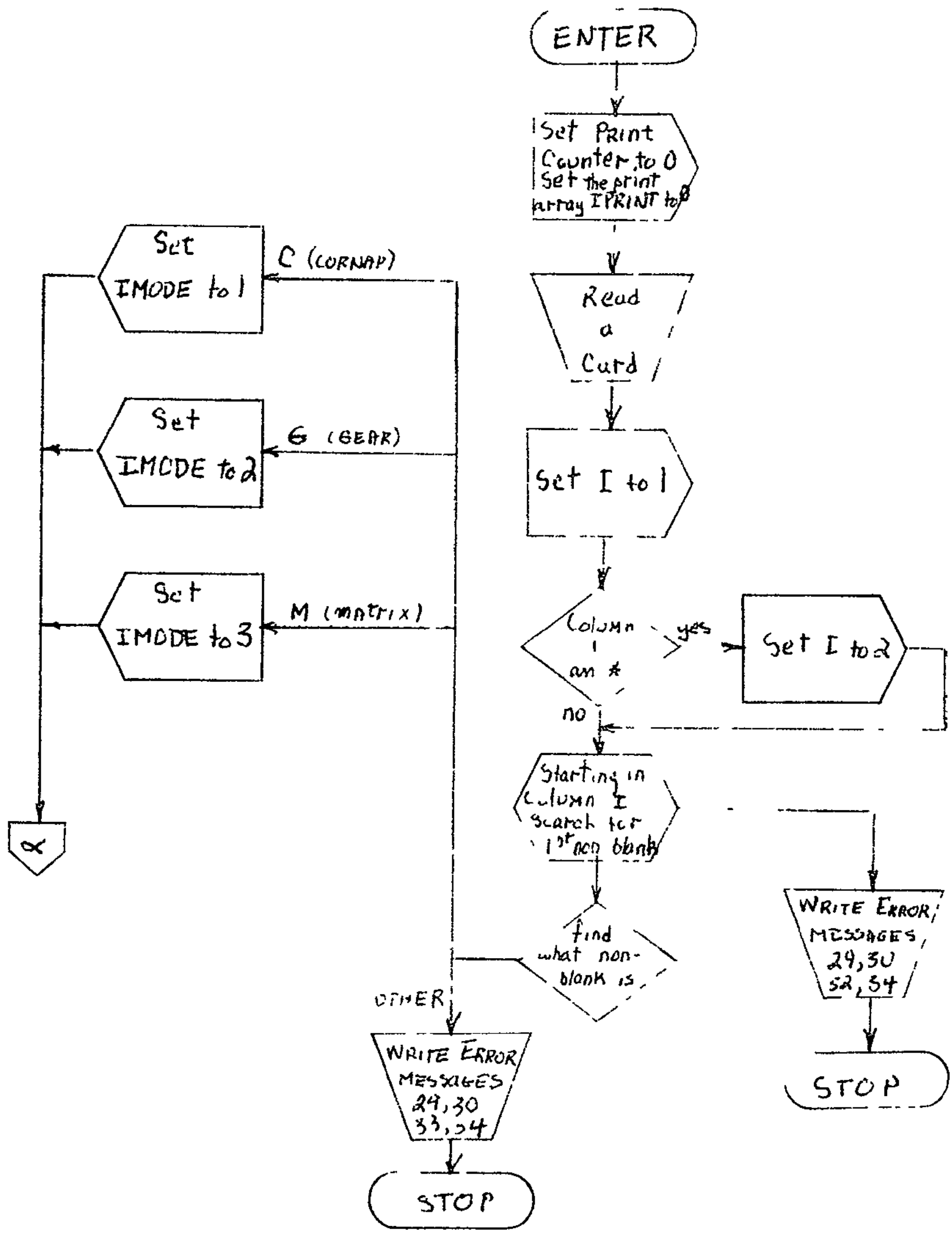
ORIGINAL PAGE IS
OF POOR QUALITY

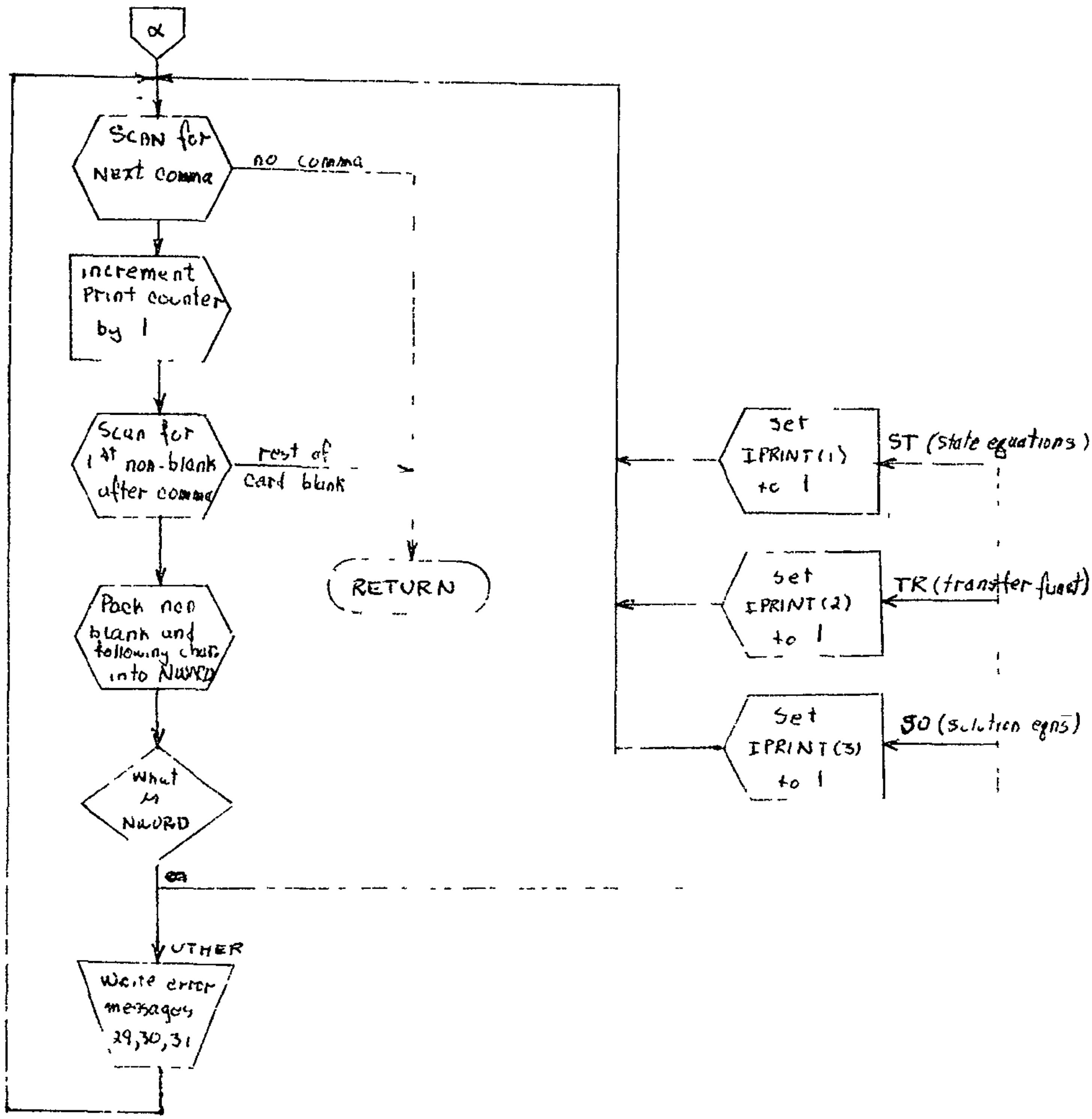

κ
Is LTYPE a "R"
yes
no
Card blank after value ?
yes
no
θ
Was resistor value = 0.0
yes
no
η
NTYPE = 3
NTYPE = 6
μ
Is LTYPE a "L"
yes
no
Card blank after value ?
yes
no
Was inductor value = 0.0 ?
no
NTYPE = 7
μ
η
non-blank a "C"
no
Was inductor value = 0.0
Write error message 62
μ
NTYPE = 8
η
Write error message 60
Set NERR to 1
γ
Is LTYPE a 'C'
Card blank after value ?
no
yes
non-blank a "T"
Was capacitor value = 0.0 ?
NTYPE = 4
μ
Write error message 69
γ
Was capacitor value = 0.0 ?
yes
NTYPE = 9
y
NTYPE = 5
μ
Write error message 60
Write error message 63
y
Set NERR to 1
γ


λ
Card after value blank ?
yes
Store 1st non-blank after value (NCTRL)
Find name after NCTRL (WORD)
Trouble finding name
δ
Is LTYPE a 'V' ?
no
yes
NTYPE - 10
NTYPE =2
μ
Is LTYPE a 'V' ?
yes
NTYPE=1
I=1
JI=NF(1)+1
CONBCH(JI,1) = WORD
NTYPE =11
I=2
JI= NF(11)+1
CONBCH(JI,2) =WORD
What is NCTRL ?
V
other
I
Write error message 64
Set NERR to 1
γ
JJ=1
JJ=2
ICON(JI,I) = JJ
μ

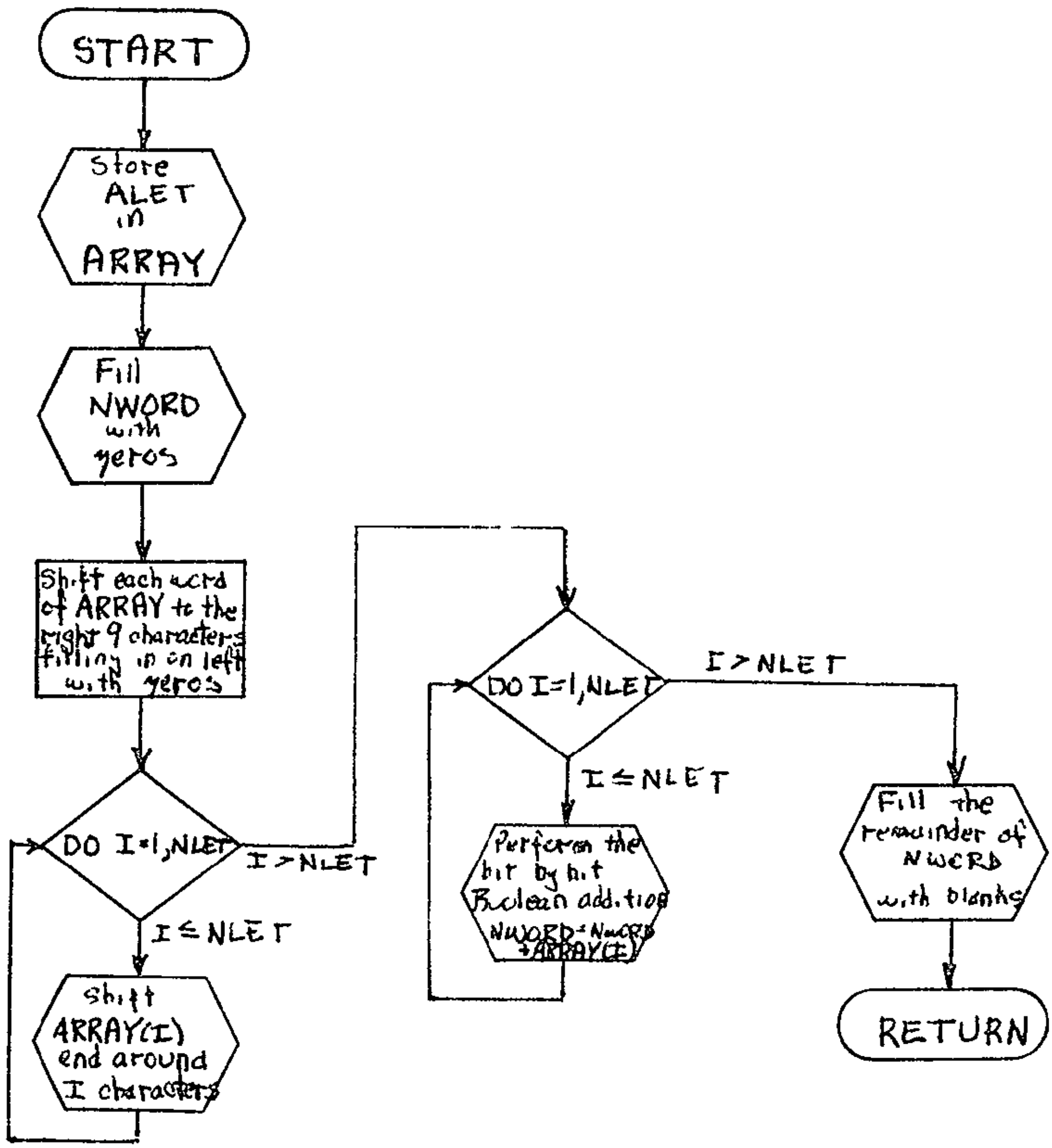

μ
increase NF(NTYPE) by 1
JI = NF(NTYPE)
Store element name (NWOR) in LIST(JI,1, NTYPE)
store element value (DEC) in LIST(JI,2, NTYPE)
store a 0 in LIST(JI,3,NTYPE)
store 1st node value(NUM1) in LIST(JI,4, NTYPE)
store 2nd node value(NUM2) in LIST(JI,5, NTYPE)
K2=NUM2+1
NODEG(K2) = NODEG(K2)+1
K1=NUM1+1
NODEG(K1) = NODEG(K1)+1
Set NODES to the largest of NODES, K1, and K2
γ
θ
Write error message
Set NERR to 1
γ
δ
what's wrong with name?
not present
> 4 characters long
Write error message 65
Write error message 55
Set NERR to 1
γ
ι
what's wrong with node?
not present
> 2 characters
> 64
Write error message 68
Write error message 56
Write error message 57
Set NERR to 1
γ
ξ
what's wrong with value?
> 15 characters
exponent > 24
Write error message 53
Write error message 54
Set NERR to 1
γ

MODE

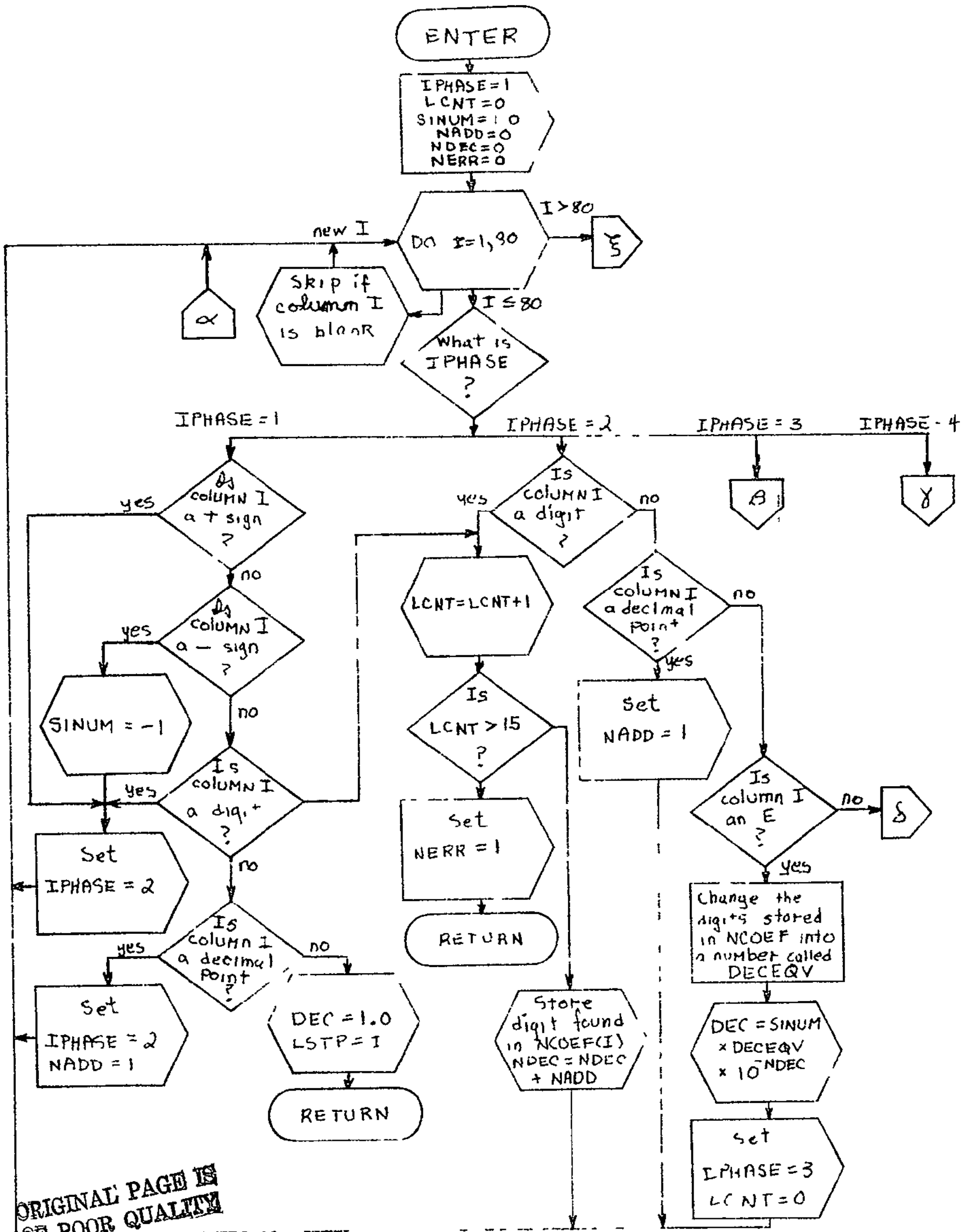

ENTER
Set Print Counter to 0 Set the print array IPRINT to 0
Read a Card
Set I to 1
Column 1 an *
yes
no
Set I to 2
Starting in column I search for 1st non blank
find what non-blank is
C (CORNAP)
G (GEAR)
M (matrix)
OTHER
Set IMODE to 1
Set IMODE to 2
Set IMODE to 3
α
WRITE ERROR MESSAGES 29,30 33,34
STOP
WRITE ERROR MESSAGES 29,30 52,54
STOP

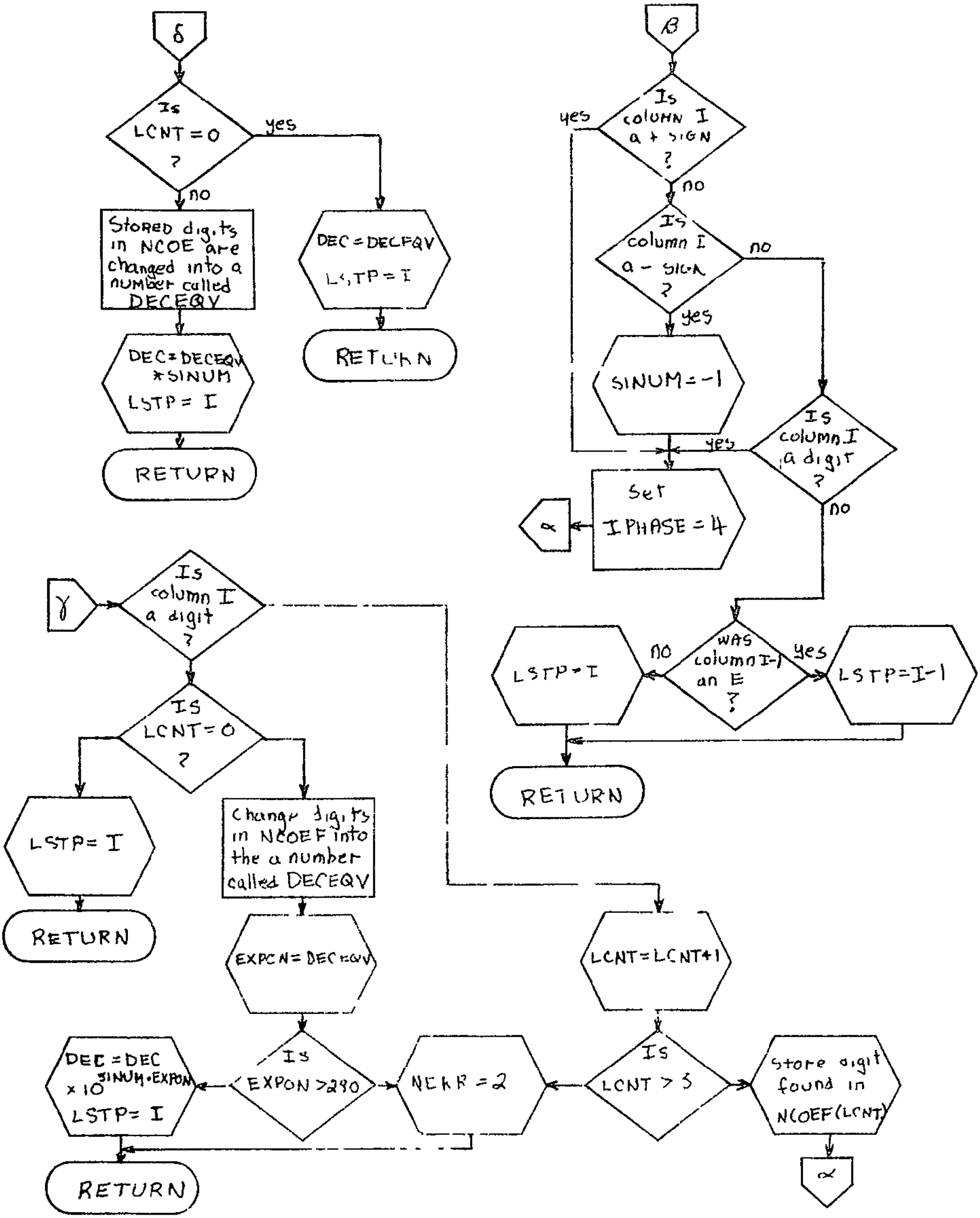

α
Scan for Next comma
no comma
increment Print counter by 1
Scan for 1st non-blank after comma
rest of card blank
RETURN
Pack non blank and following chars into NWORD
What is NWORD
OTHER
Write error messages 29,30,31
Set IPRINT(1) to 1
ST (state equations)
Set IPRINT(2) to 1
TR (transfer funct)
Set IPRINT(3) to 1
SO (solution eqns)

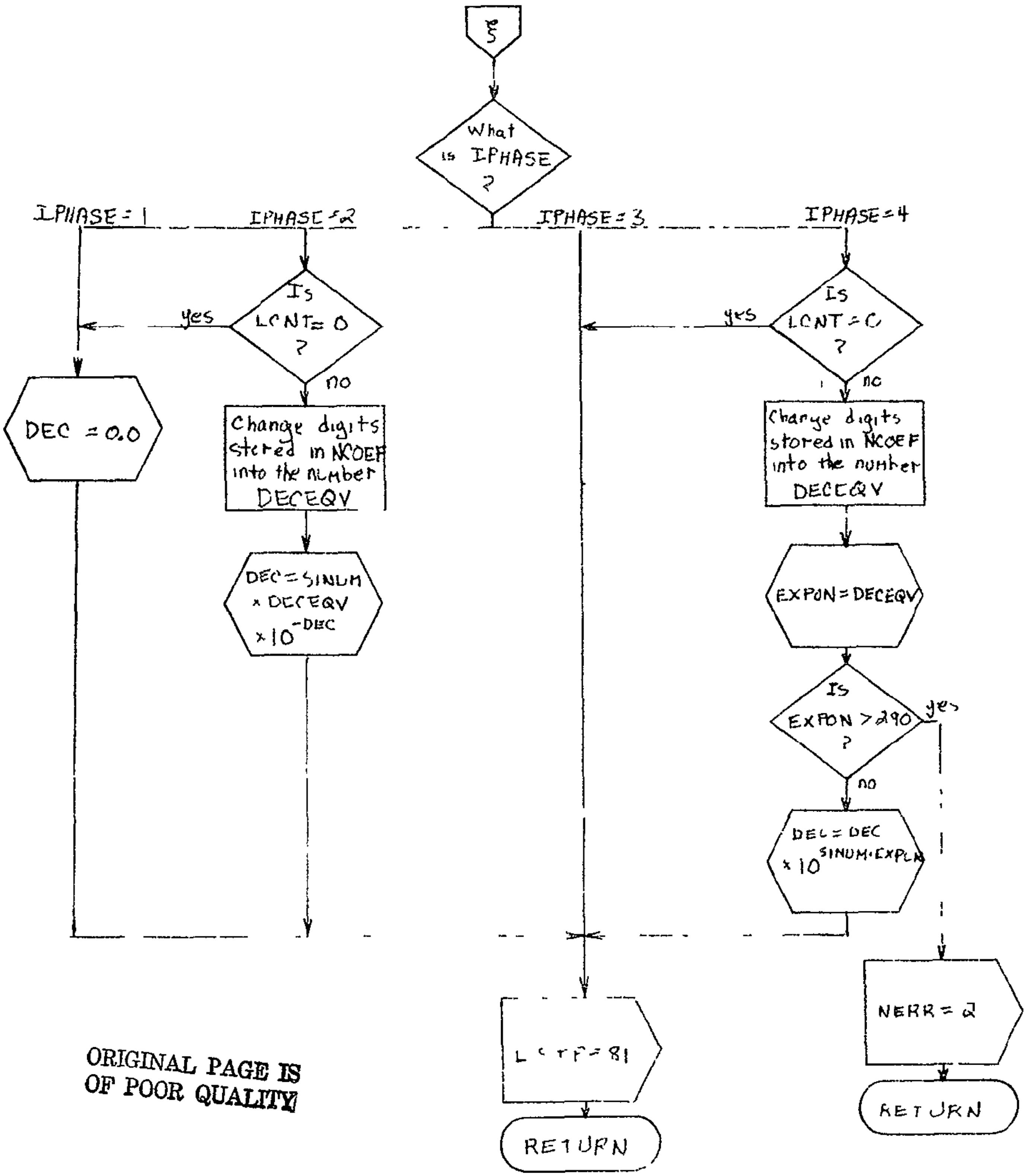
APACK


START
Store ALET in ARRAY
Fill NWORD with zeros
Shift each word of ARRAY to the right 9 characters filling in on left with zeros
DO I=1,NLET
I ≤ NLET
I > NLET
Shift ARRAY(I) end around I characters
DO I=1,NLET
I ≤ NLET
I > NLET
Perform the bit by bit Boolean addition NWORD=NWORD +ARRAY(I)
Fill the remainder of NWORD with blanks
RETURN

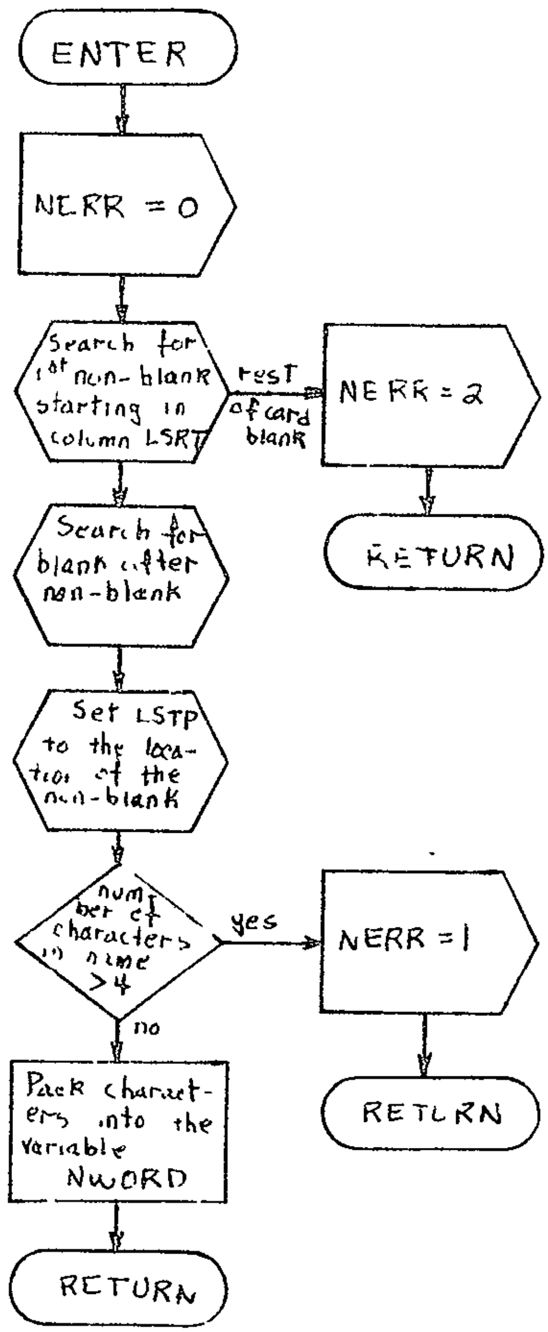
ATODEC


ENTER
IPHASE=1
LCNT=0
SINUM=1.0
NADD=0
NDEC=0
NERR=0
Do I=1,80
I>80
new I
Skip if column I is blank
I≤80
What is IPHASE ?
IPHASE=1
IPHASE=2
IPHASE=3
IPHASE=4
Is column I a + sign ?
yes
no
Is column I a − sign ?
SINUM = −1
Is column I a digit ?
Set IPHASE = 2
Is column I a decimal point ?
Set IPHASE = 2 NADD = 1
DEC = 1.0 LSTP = I
RETURN
Is column I a digit ?
LCNT=LCNT+1
Is LCNT > 15 ?
Set NERR = 1
RETURN
Store digit found in NCOEF(I) NDEC = NDEC + NADD
Is column I a decimal point ?
Set NADD = 1
Is column I an E ?
Change the digits stored in NCOEF into a number called DECEQV
DEC = SINUM × DECEQV × 10^-NDEC
Set IPHASE=3 LCNT=0

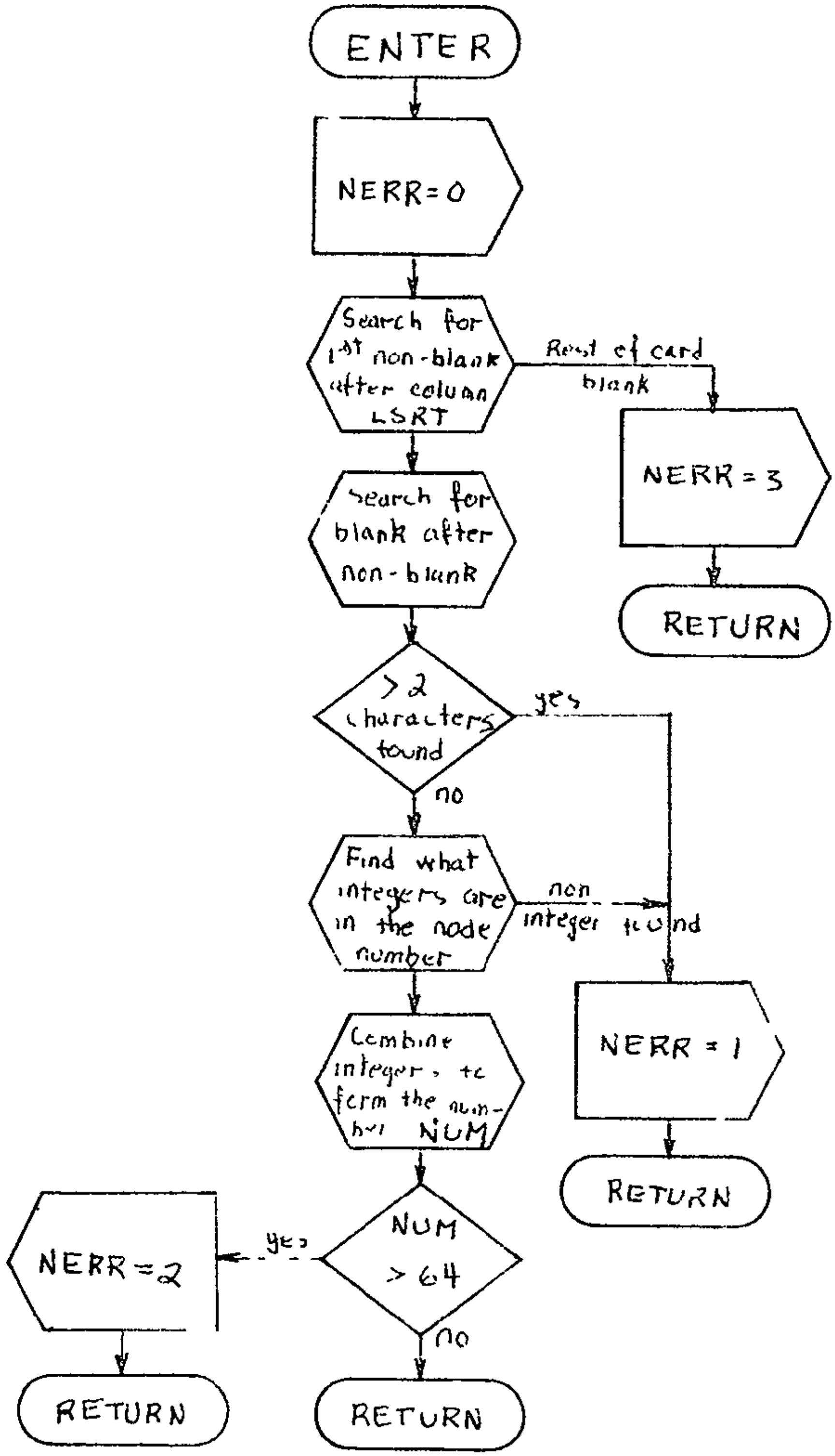

δ
Is LCNT = 0 ?
yes
no
Stored digits in NCOE are changed into a number called DECEQV
DEC = DECEQV * SINUM
LSTP = I
RETURN
DEC = DECEQV
LSTP = I
RETURN
β
Is column I a + SIGN ?
yes
no
Is column I a - SIGN ?
no
yes
SINUM = -1
Is column I a digit ?
yes
no
Set IPHASE = 4
α
WAS column I-1 an E ?
no
yes
LSTP = I
LSTP = I-1
RETURN
γ
Is column I a digit ?
Is LCNT = 0 ?
LSTP = I
RETURN
Change digits in NCOEF into the a number called DECEQV
EXPON = DECEQV
Is EXPON > 290
NERR = 2
DEC = DEC × 10 SINUM·EXPON
LSTP = I
RETURN
LCNT = LCNT+1
Is LCNT > 5
Store digit found in NCOEF(LCNT)
α


ξ
What is IPHASE ?
IPHASE = 1
IPHASE = 2
IPHASE = 3
IPHASE = 4
Is LCNT = 0 ?
yes
no
DEC = 0.0
Change digits stored in NCOEF into the number DECEQV
DEC = SINUM × DECEQV × 10^-DEC
Is LCNT = 0 ?
yes
no
Change digits stored in NCOEF into the number DECEQV
EXPON = DECEQV
Is EXPON > 290 ?
yes
no
DEC = DEC × 10^SINUM·EXPON
LCTF = 81
RETURN
NERR = 2
RETURN

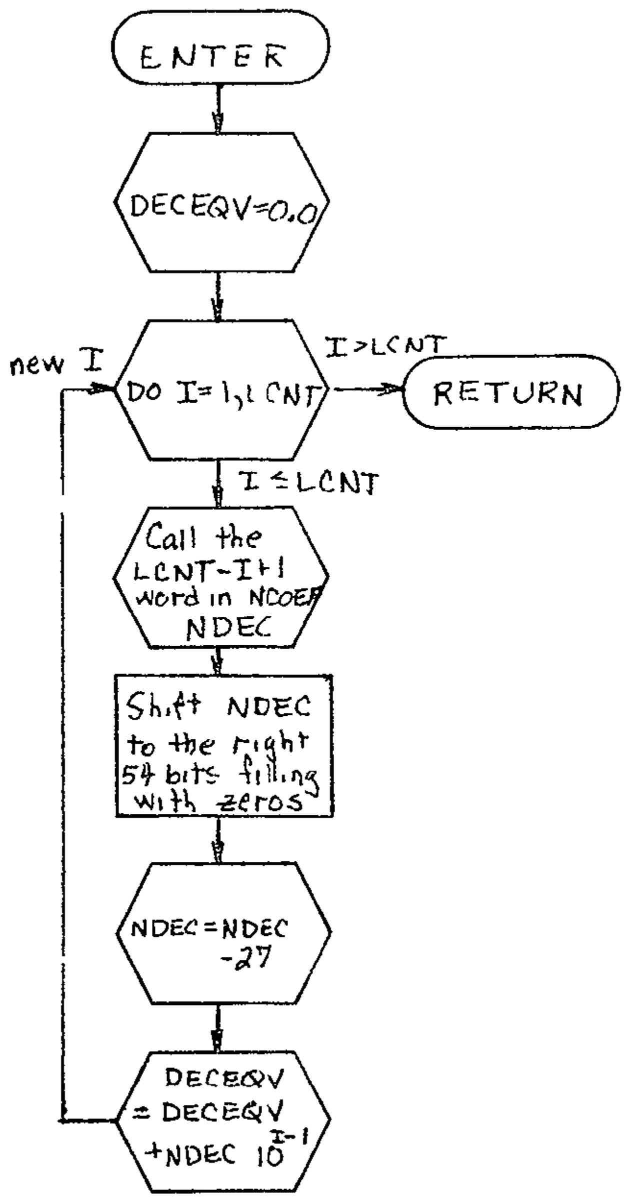
NAME


ENTER
NERR = 0
Search for 1st non-blank starting in column LSRT
rest of card blank
NERR = 2
RETURN
Search for blank after non-blank
Set LSTP to the location of the non-blank
number of characters in name > 4
yes
NERR = 1
RETURN
no
Pack characters into the variable NWORD
RETURN

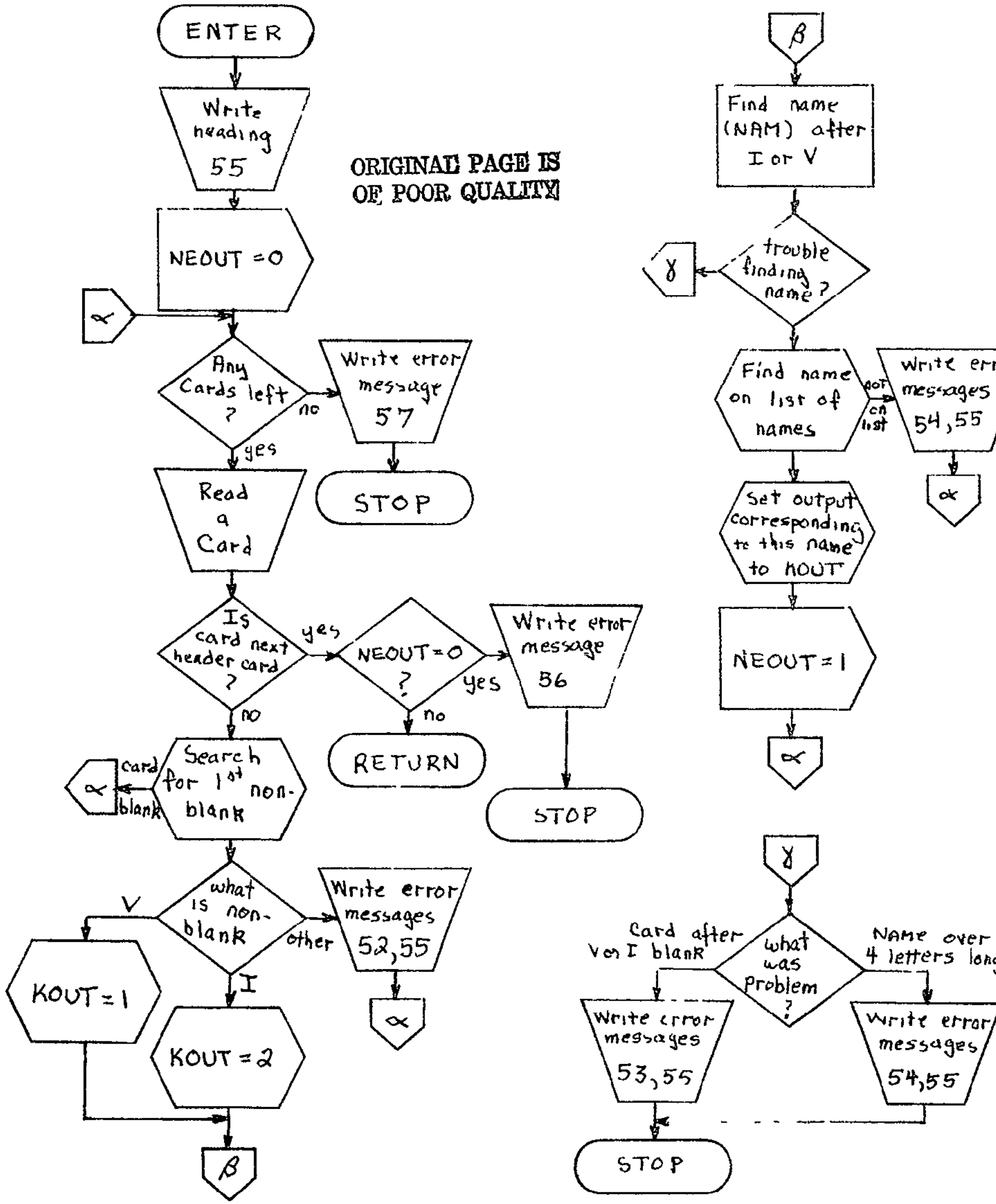
NODE


ENTER
NERR=0
Search for 1st non-blank after column LSRT
Rest of card blank
NERR=3
RETURN
Search for blank after non-blank
>2 characters found
yes
no
Find what integers are in the node number
non integer found
NERR=1
RETURN
Combine integers to form the number NUM
NUM >64
yes
NERR=2
RETURN
no
RETURN

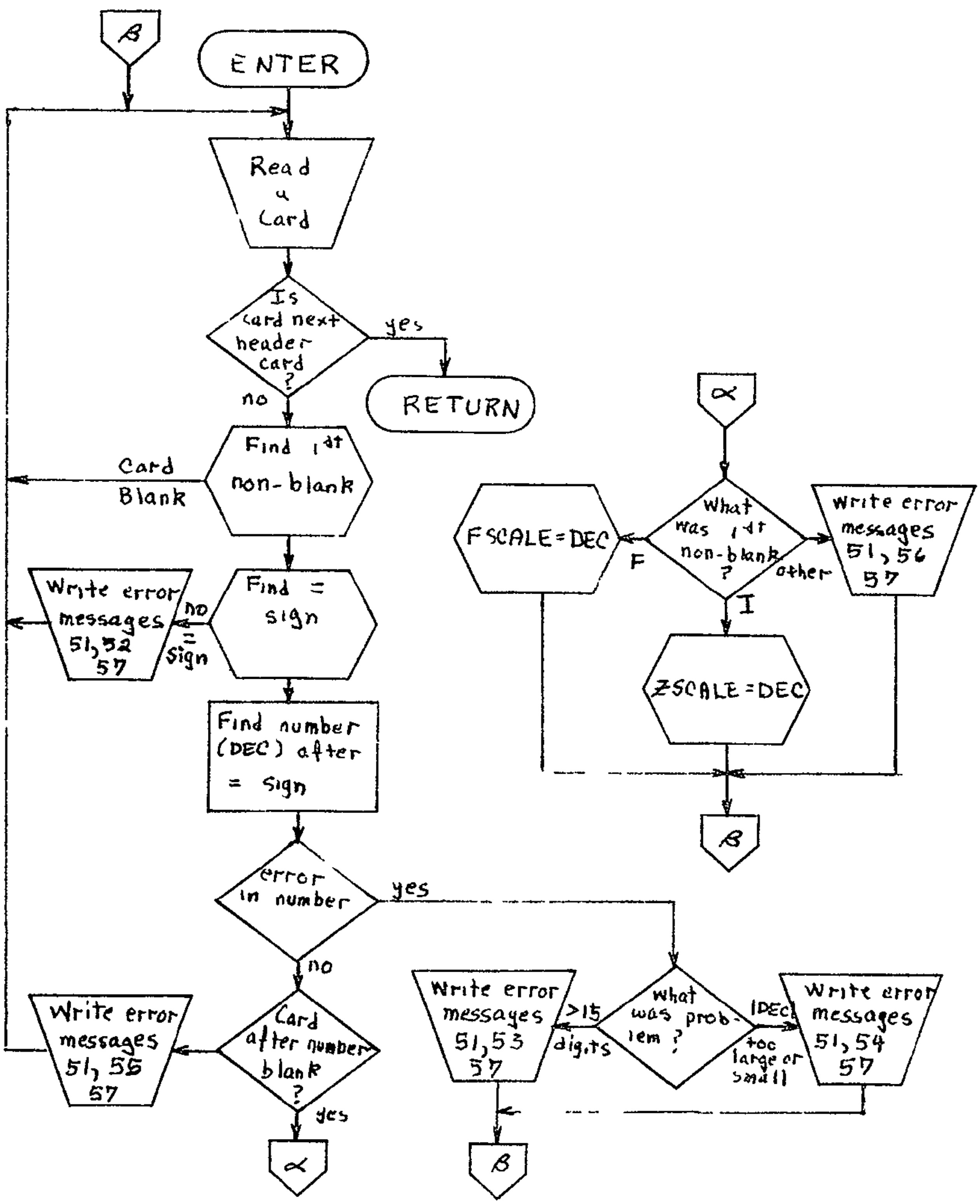
DECEQV


ENTER
DECEQV=0.0
new I
DO I=1,LCNT
I>LCNT
RETURN
I≤LCNT
Call the LCNT-I+1 word in NCOEF NDEC
Shift NDEC to the right 54 bits filling with zeros
NDEC=NDEC -27
DECEQV = DECEQV +NDEC 10^I-1

OUTPUT

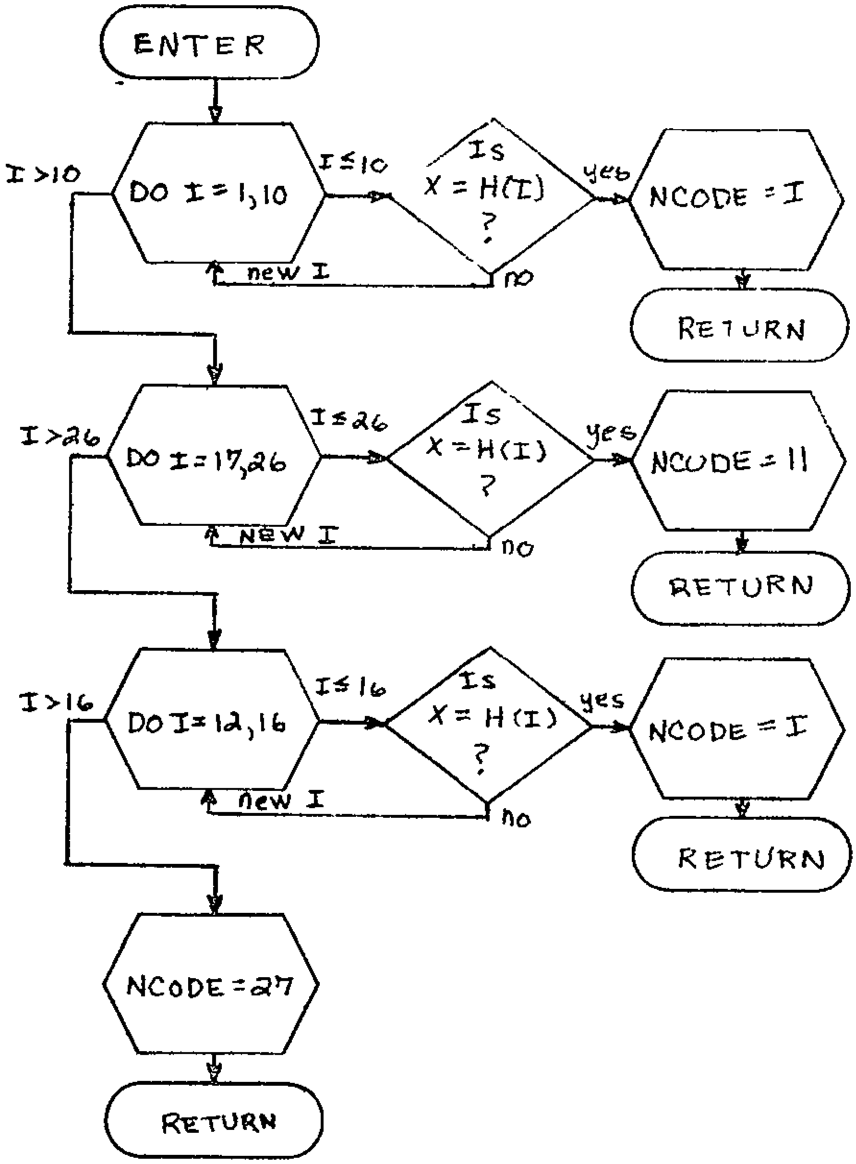

ENTER
Write heading 55
NEOUT = 0
Any Cards left ?
no
yes
Write error message 57
STOP
Read a Card
Is card next header card ?
yes
no
NEOUT = 0 ?
yes
no
Write error message 56
STOP
RETURN
Search for 1st non-blank
card blank
what is non-blank
V
I
other
Write error messages 52,55
KOUT = 1
KOUT = 2
Find name (NAM) after I or V
trouble finding name ?
Find name on list of names
not on list
Write error messages 54,55
Set output corresponding to this name to KOUT
NEOUT = 1
what was Problem ?
Card after V or I blank
Name over 4 letters long
Write error messages 53,55
Write error messages 54,55
STOP

ZFSCAL



β
ENTER
Read a Card
Is card next header card ?
yes
RETURN
no
Find 1st non-blank
Card Blank
Find = sign
no = sign
Write error messages 51, 52 57
Find number (DEC) after = sign
error in number
yes
no
Card after number blank ?
yes
α
Write error messages 51, 55 57
What was problem ?
>15 digits
Write error messages 51, 53 57
β
1DEC too large or small
Write error messages 51, 54 57
α
What was 1st non-blank ?
F
FSCALE=DEC
I
ZSCALE=DEC
other
Write error messages 51, 56 57
β

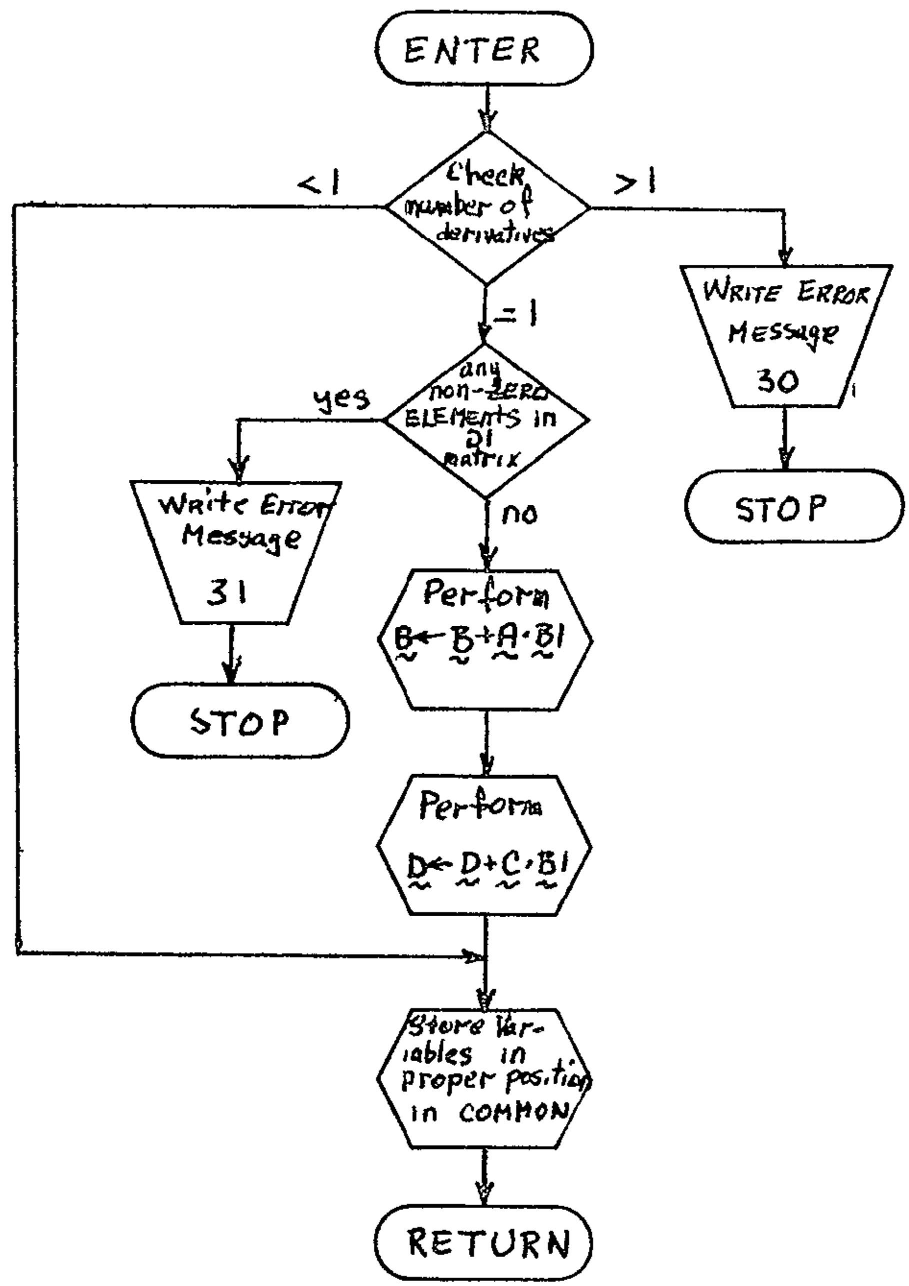
NCODE



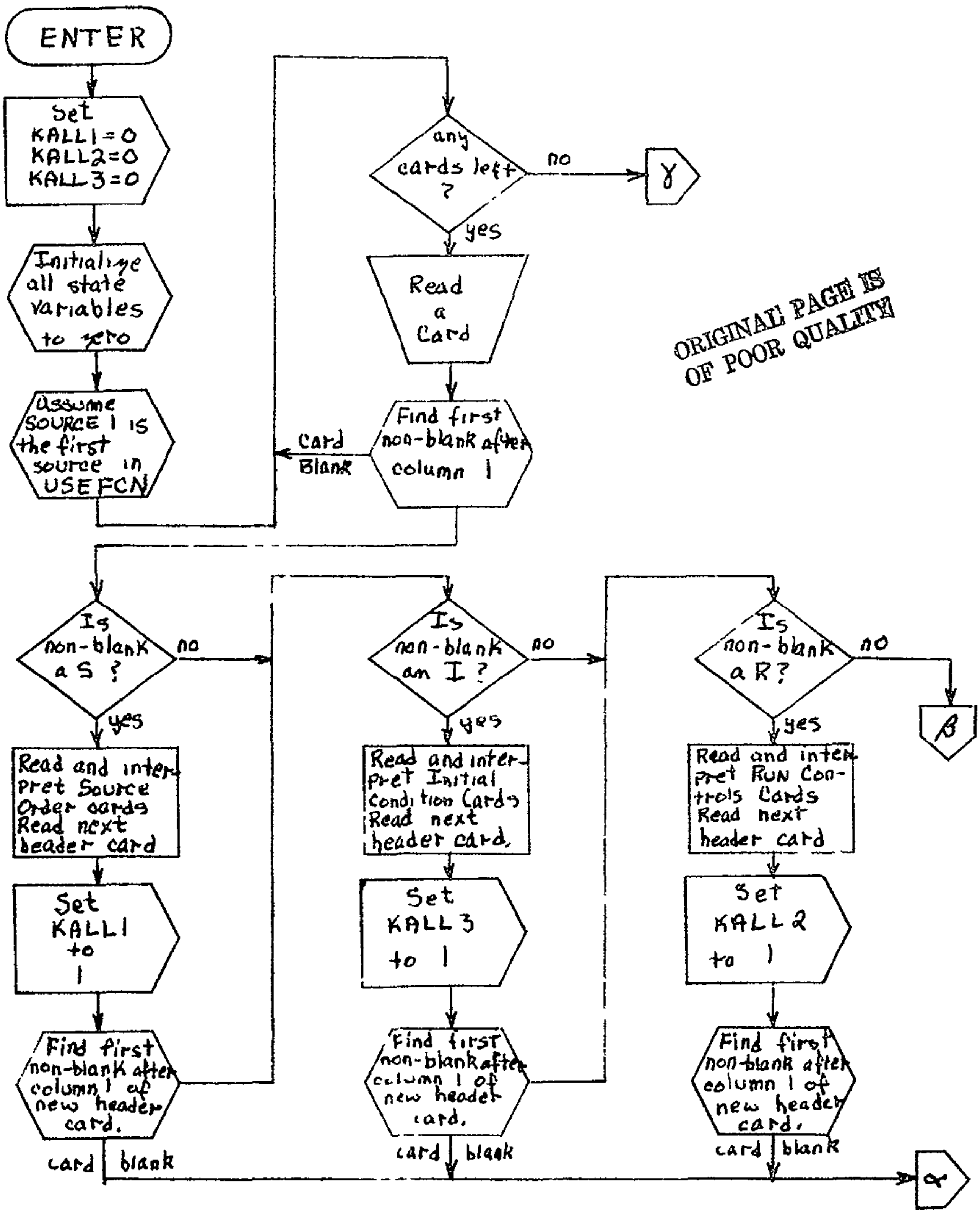
here the array H contains

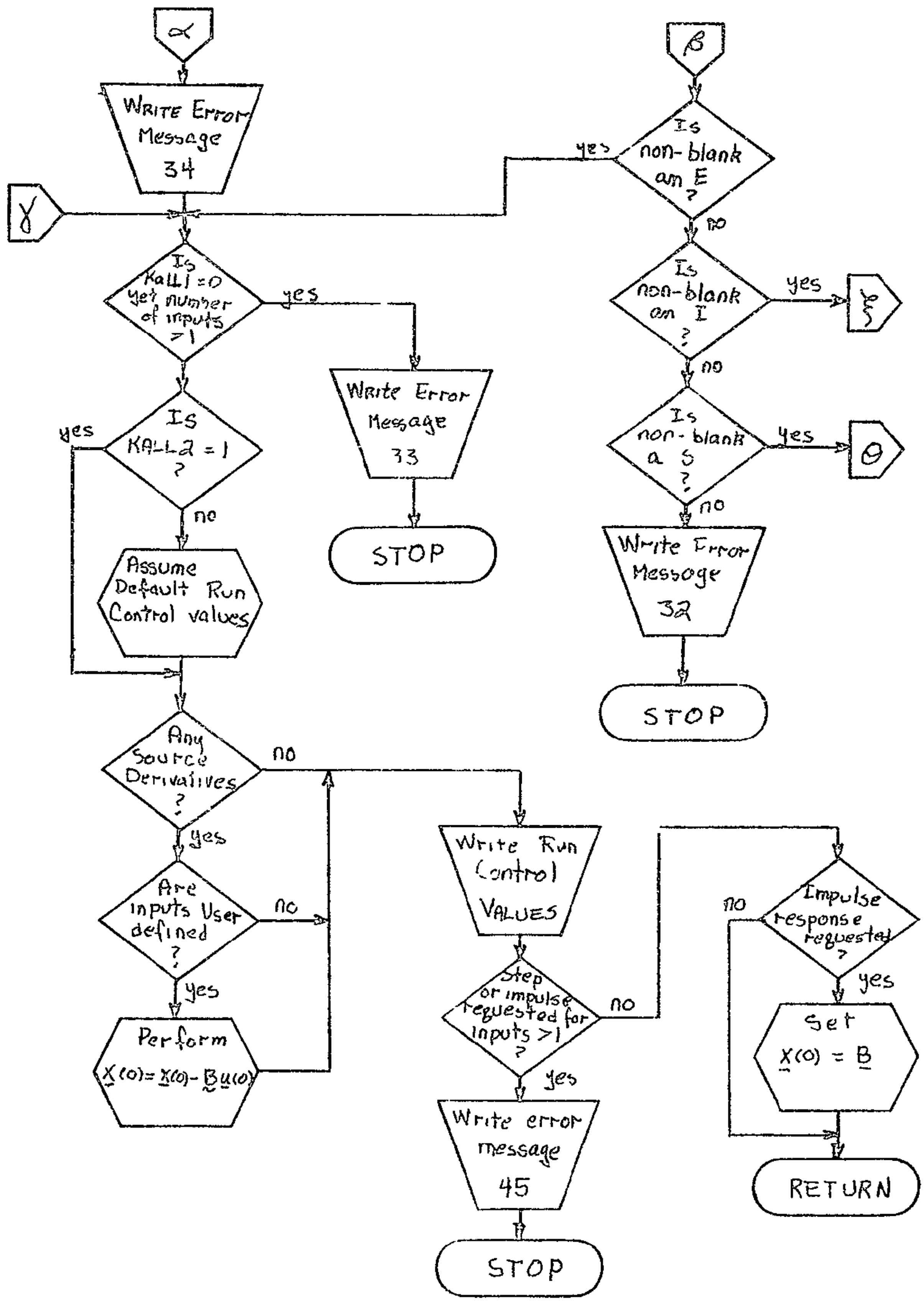
H(1)	1HV
H(2)	1HC
H(3)	1HR
H(4)	1HL
H(5)	1HI
H(6)	1H
H(7)	1H+
H(8)	1H-
H(9)	1HK
H(10)	1HM
H(11)	1H+
H(12)	1H(
H(13)	1H)
H(14)	1H
H(15)	1HE
H(16)	1H*
H(17)	1H1
H(18)	1H2
H(19)	1H3
H(20)	1H4
H(21)	1H5
H(22)	1H6
H(23)	1H7
H(24)	1H8
H(25)	1H9
H(26)	1H0

COMPAK

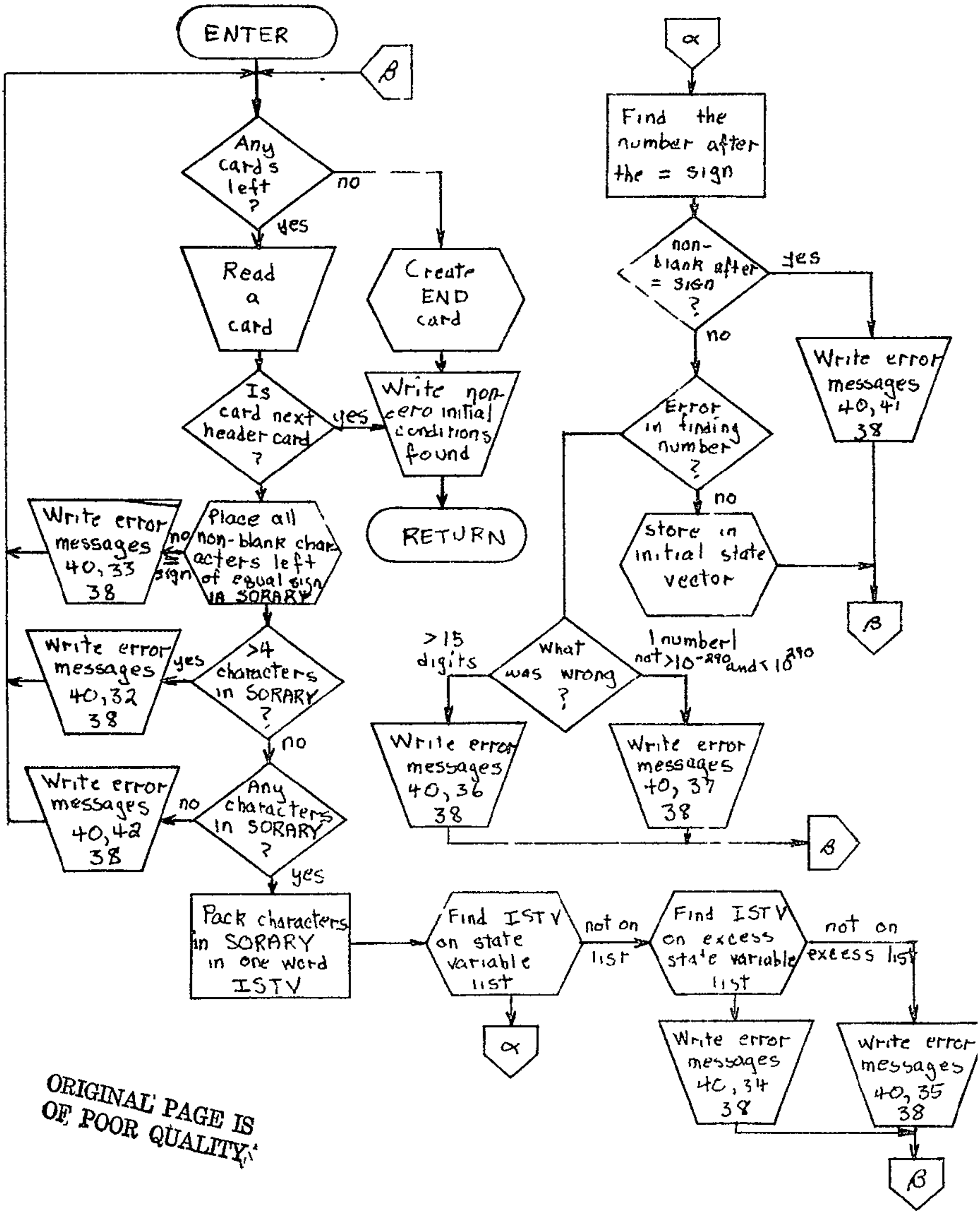

ENTER
Check number of derivatives
<1
>1
=1
WRITE ERROR Message 30
STOP
any non-ZERO ELEMENTS in BI matrix
yes
no
Write Error Message 31
STOP
Perform B ← B + A · BI
Perform D ← D + C · BI
Store Variables in proper position in COMMON
RETURN

CROLGR


ENTER
Set KALL1=0 KALL2=0 KALL3=0
Initialize all state variables to zero
Assume SOURCE 1 is the first source in USEFCN
any cards left ?
no
yes
Read a Card
Find first non-blank after column 1
card Blank
Is non-blank a S ?
no
yes
Read and interpret Source Order cards Read next header card
Set KALL1 to 1
Find first non-blank after column 1 of new header card.
card blank
Is non-blank an I ?
no
yes
Read and interpret Initial Condition Cards Read next header card.
Set KALL3 to 1
Find first non-blank after column 1 of new header card.
card blank
Is non-blank a R?
no
yes
Read and interpret Run Controls Cards Read next header card
Set KALL2 to 1
Find first non-blank after column 1 of new header card.
card blank
γ
β
α

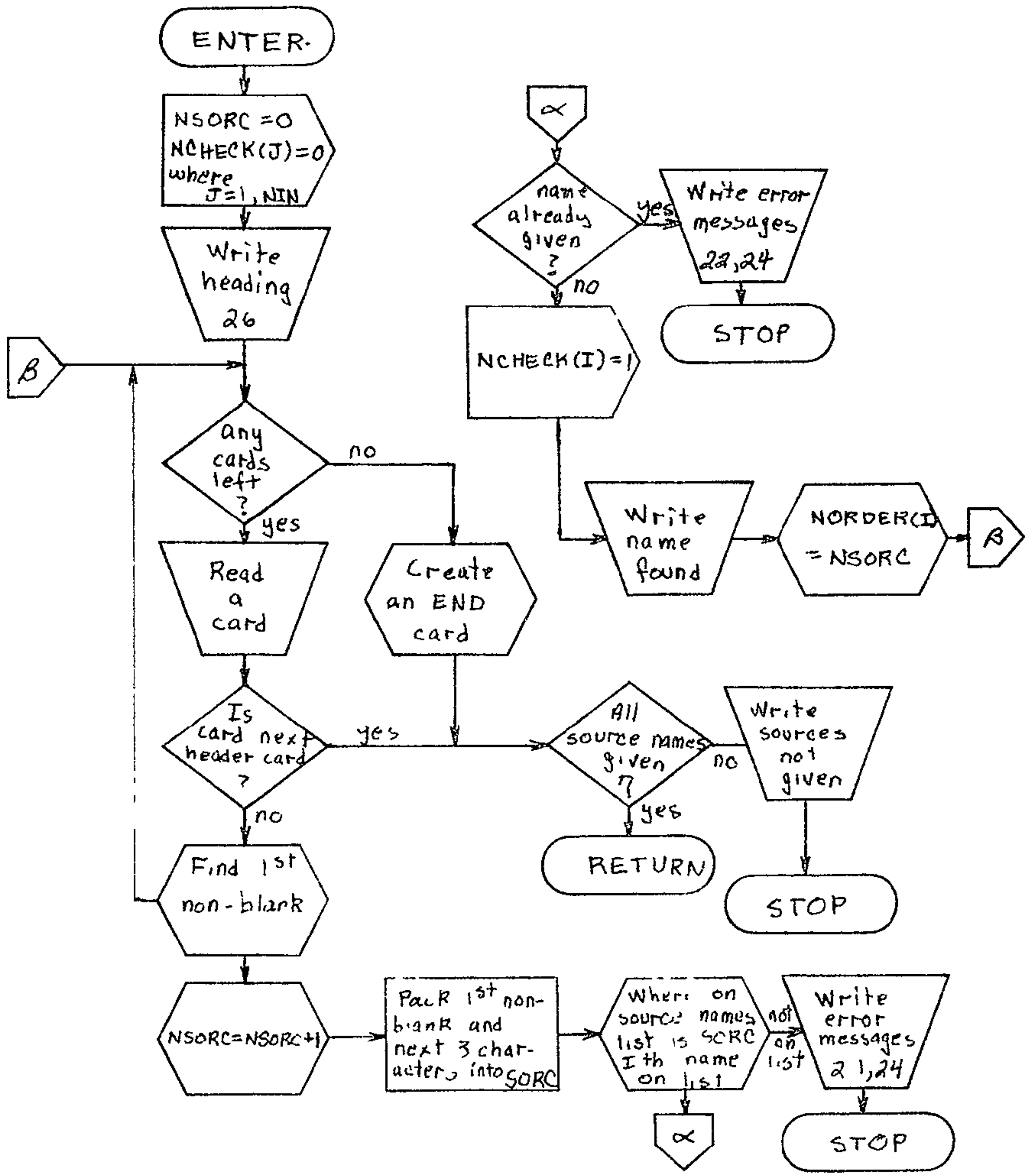

α
Write Error Message 34
β
yes
Is non-blank an E ?
no
γ
Is Kall1 = 0 yet number of inputs > 1
yes
Is non-blank an I ?
yes
ξ
no
Write Error Message 33
yes
Is KALL2 = 1 ?
Is non-blank a S ?
yes
θ
no
no
STOP
Assume Default Run Control values
Write Error Message 32
STOP
Any Source Derivatives ?
no
yes
Write Run Control Values
Are inputs User defined ?
no
no
Impulse response requested ?
yes
Step or impulse requested for inputs > 1 ?
no
yes
Perform
X(0) = X(0) - B u(0)
Set
X(0) = B
yes
Write error message 45
RETURN
STOP

ICONSV

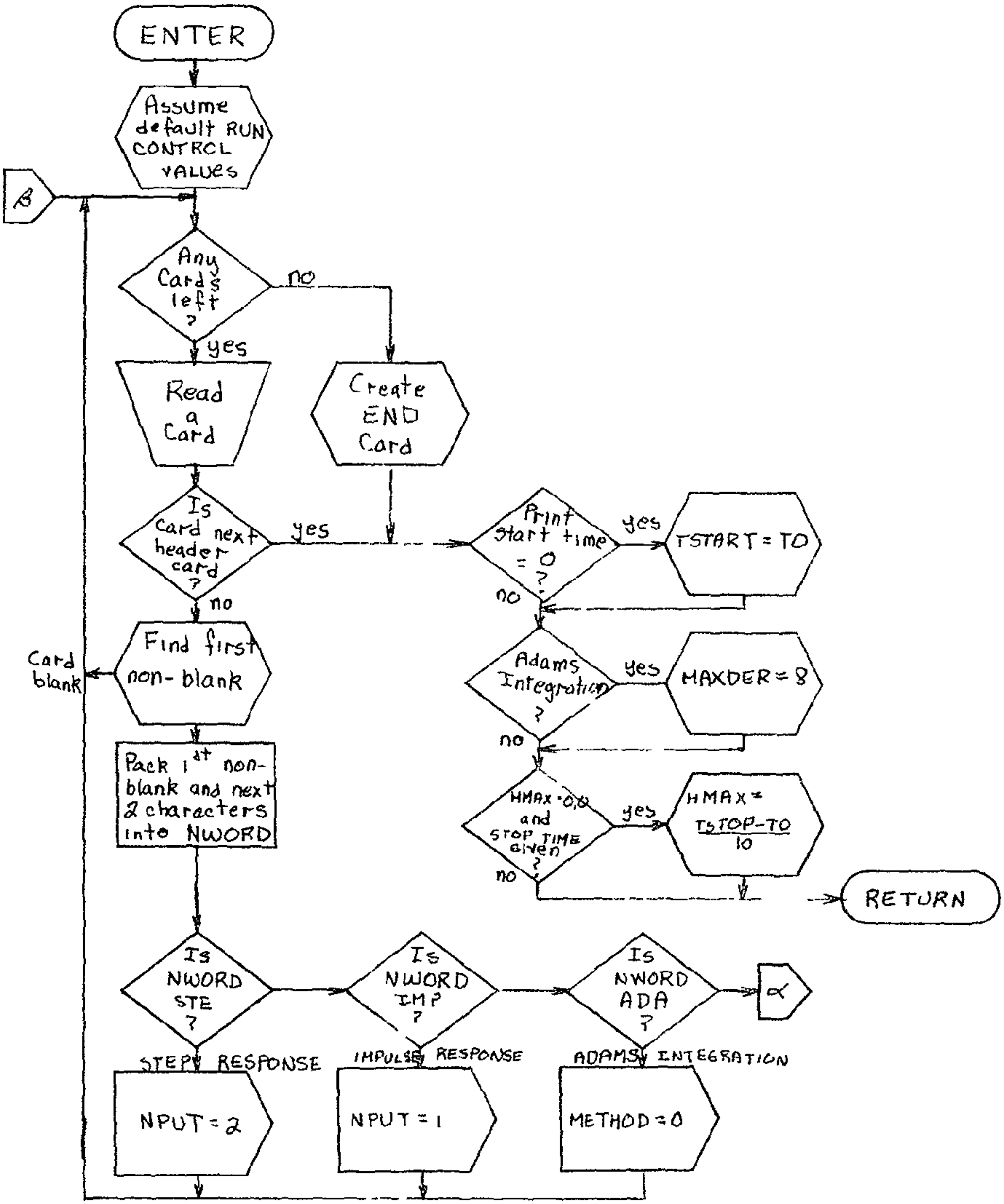


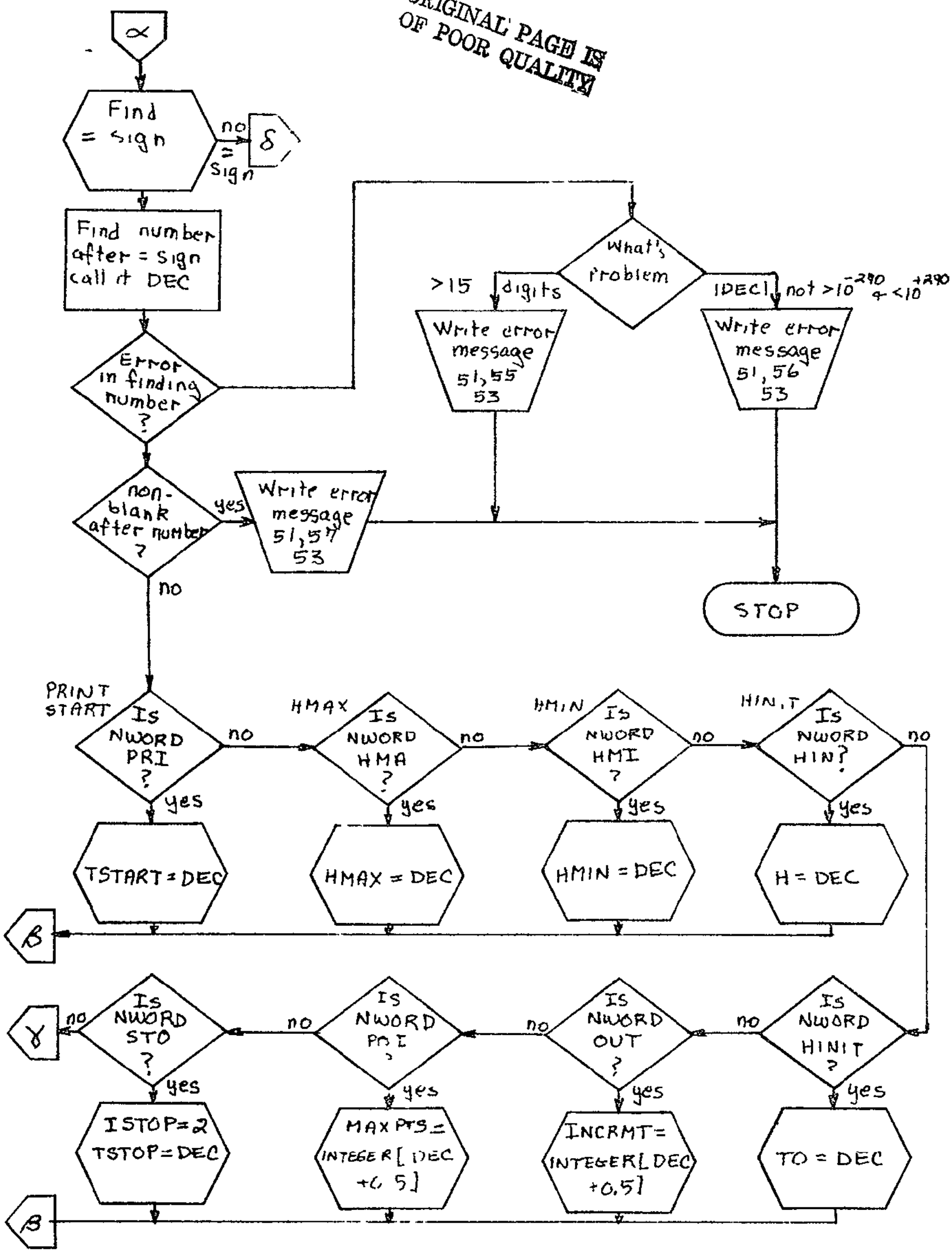
ORIGINAL PAGE IS OF POOR QUALITY

SRCORD

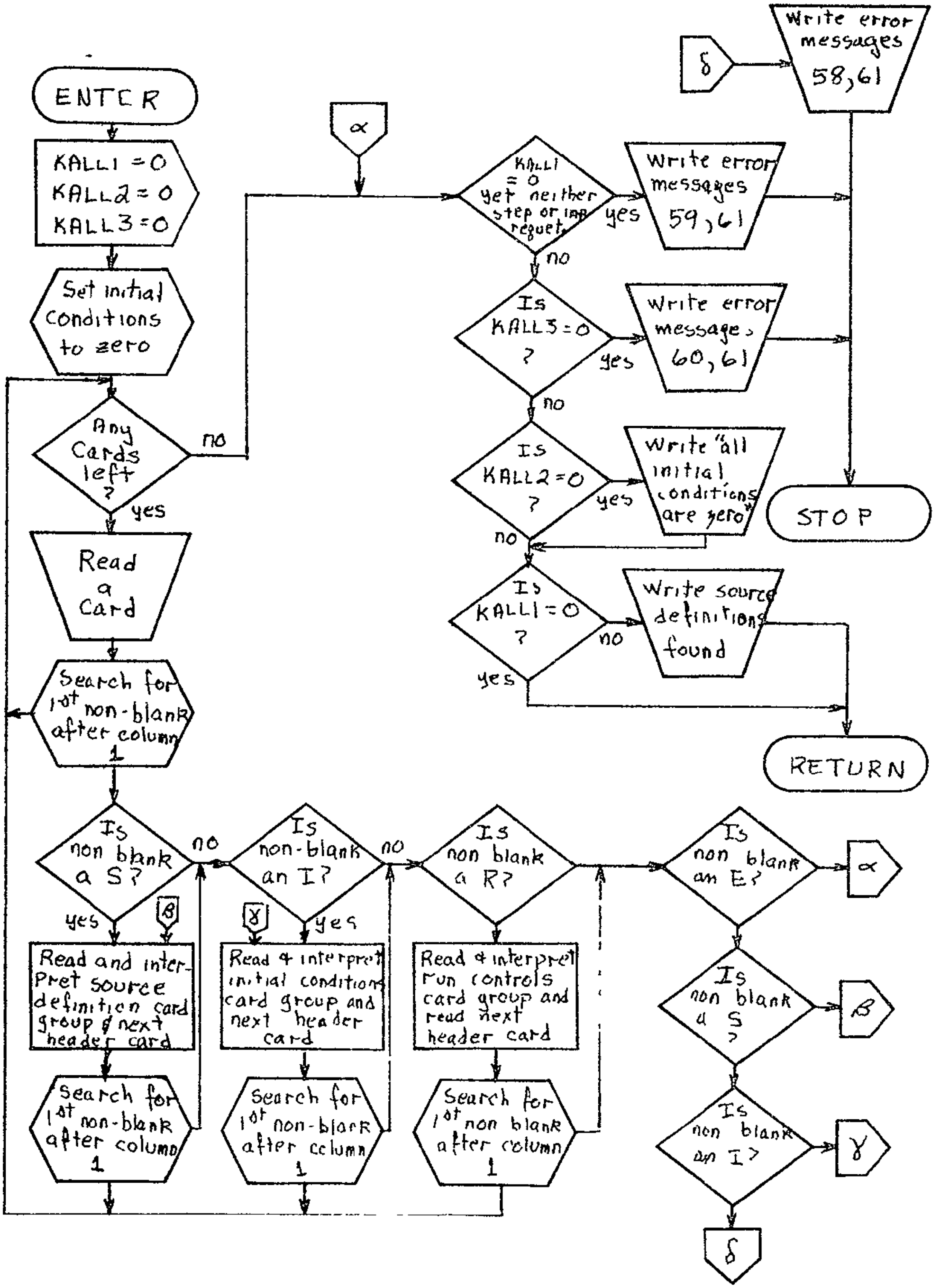

ENTER
NSORC = 0
NCHECK(J)=0
where J=1, NIN
Write heading 26
β
any cards left ?
no
yes
Read a card
Create an END card
Is card next header card ?
yes
no
Find 1st non-blank
NSORC=NSORC+1
Pack 1st non-blank and next 3 characters into SORC
Where on source names list is SORC Ith name on list
not on list
Write error messages 21,24
STOP
α
All source names given ?
no
yes
Write sources not given
STOP
RETURN
α
name already given ?
yes
no
Write error messages 22,24
STOP
NCHECK(I)=1
Write name found
NORDER(I) = NSORC
β

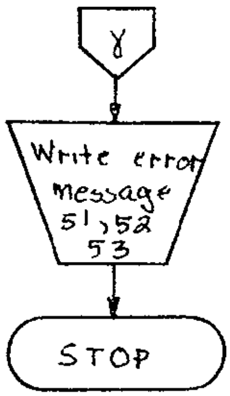
RUNCTL

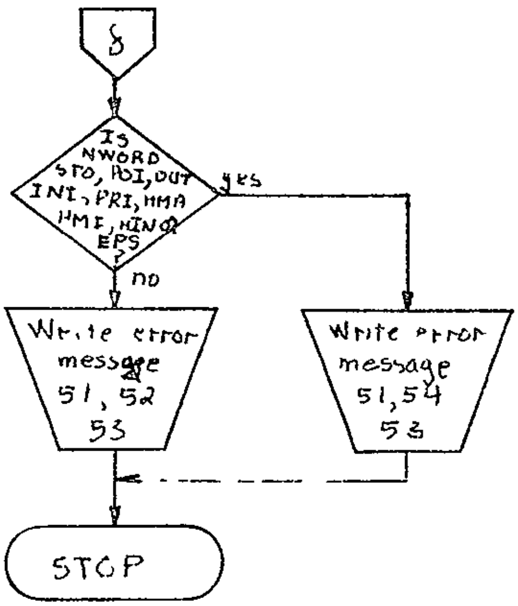

ENTER
Assume default RUN CONTROL VALUES
β
Any Cards left ?
no
yes
Read a Card
Create END Card
Is Card next header card ?
yes
no
Print start time = 0 ?
yes
no
TSTART = TO
Adams Integration ?
yes
no
MAXDER = 8
HMAX = 0.0 and STOP TIME given ?
yes
no
HMAX = TSTOP-TO / 10
RETURN
Card blank
Find first non-blank
Pack 1st non-blank and next 2 characters into NWORD
Is NWORD STE ?
Is NWORD IMP ?
Is NWORD ADA ?
α
STEP RESPONSE
IMPULSE RESPONSE
ADAMS INTEGRATION
NPUT = 2
NPUT = 1
METHOD = 0


α
Find = sign
no = sign
δ
Find number after = sign call it DEC
Error in finding number ?
What's problem
>15 digits
|DEC| not >10^-290 & <10^+290
Write error message 51,55 53
Write error message 51,56 53
non-blank after number ?
yes
Write error message 51,57 53
no
STOP
PRINT START
Is NWORD PRI ?
HMAX
Is NWORD HMA ?
HMIN
Is NWORD HMI ?
HINIT
Is NWORD HIN?
no
yes
TSTART = DEC
HMAX = DEC
HMIN = DEC
H = DEC
β
Is NWORD HINIT ?
Is NWORD OUT ?
Is NWORD PRI ?
Is NWORD STO ?
γ
TO = DEC
INCRMT = INTEGER[DEC +0.5]
MAXPTS = INTEGER[DEC +0.5]
ISTOP=2 TSTOP=DEC
β

CROLMX

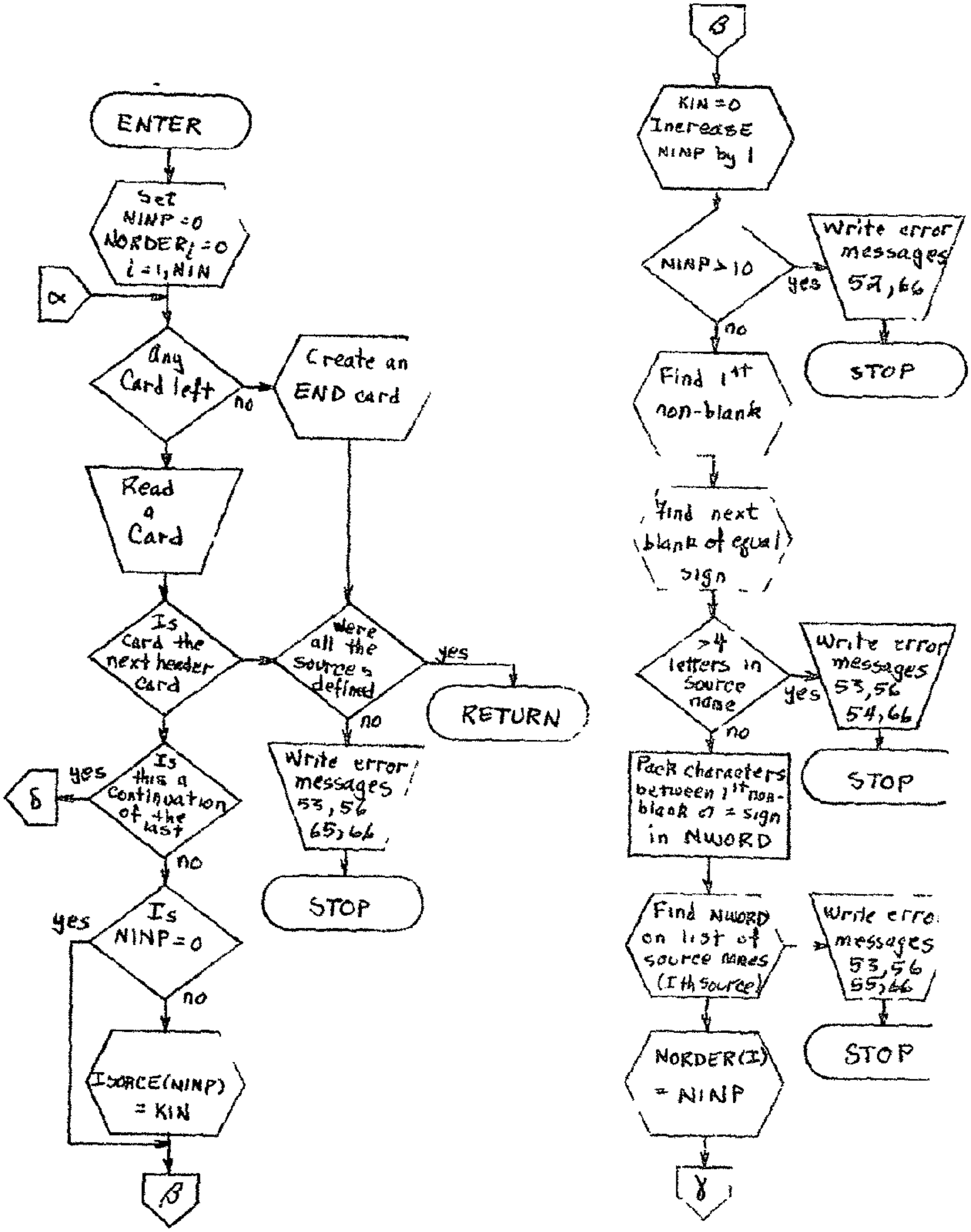

ENTER
KALL1 = 0
KALL2 = 0
KALL3 = 0
Set initial conditions to zero
Any Cards left ?
no
yes
Read a Card
Search for 1st non-blank after column 1
Is non blank a S?
yes
no
Read and interpret source definition card group & next header card
Search for 1st non-blank after column 1
Is non-blank an I?
yes
no
Read & interpret initial conditions card group and next header card
Search for 1st non-blank after column 1
Is non blank a R?
Read & interpret run controls card group and read next header card
Search for 1st non blank after column 1
Is non blank an E?
Is non blank a S ?
Is non blank an I?
α
β
γ
δ
KALL1 = 0 yet neither step or imp request
yes
no
Write error messages 59, 61
Is KALL3 = 0 ?
Write error messages 60, 61
Is KALL2 = 0 ?
Write "all initial conditions are zero"
Is KALL1 = 0 ?
Write source definitions found
Write error messages 58, 61
STOP
RETURN

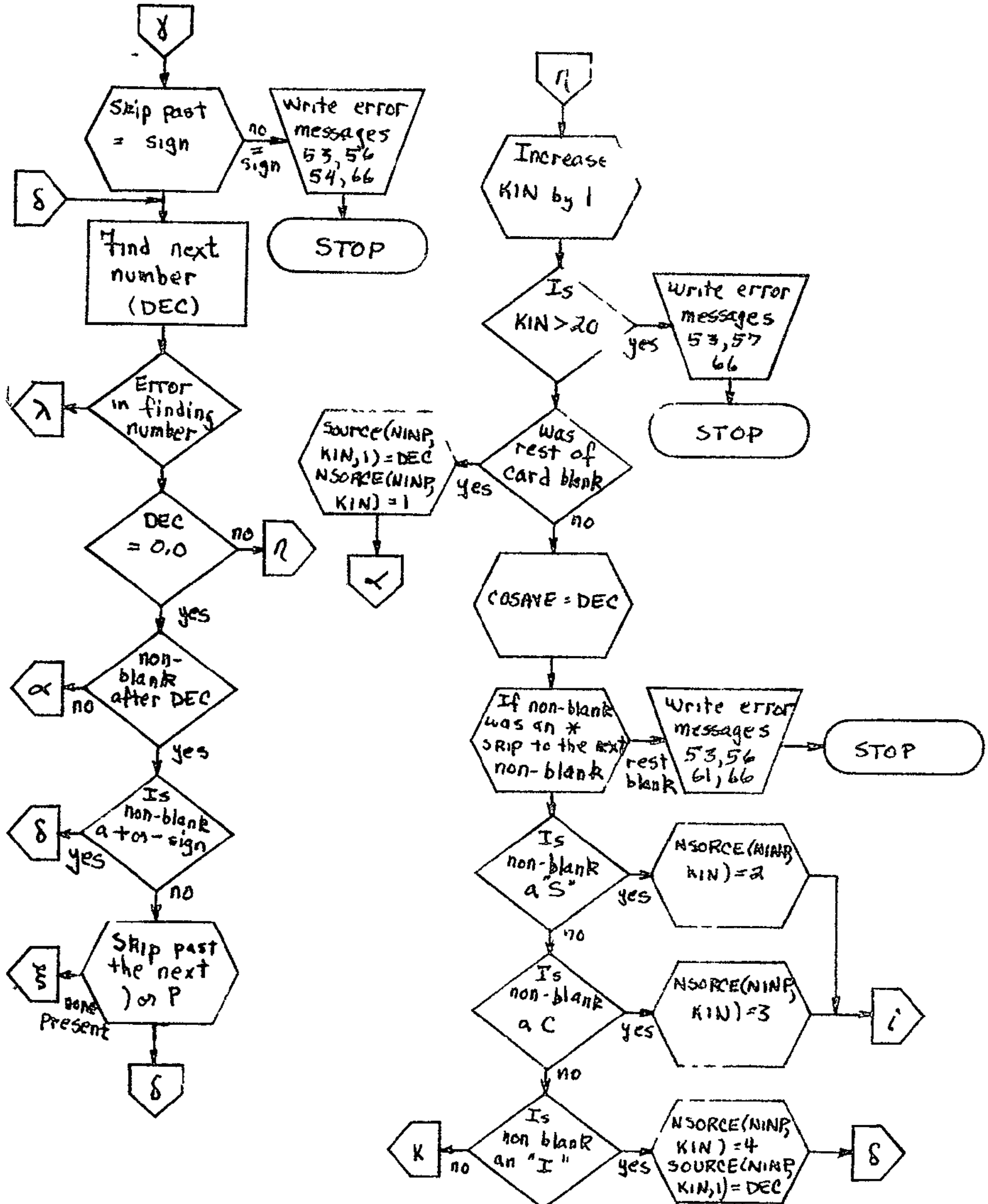

γ
Write error message 51, 52 53
STOP

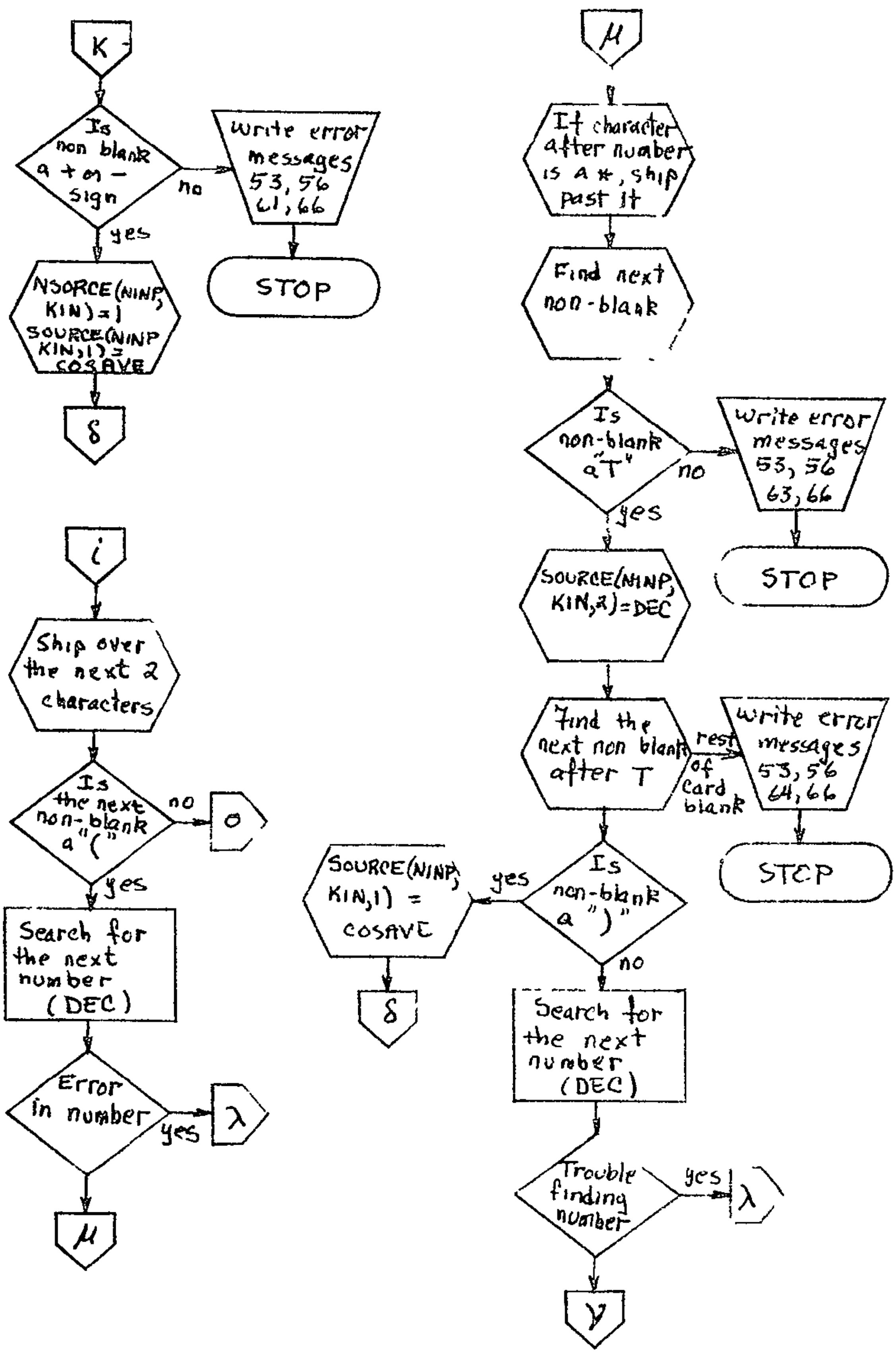

δ
IS NWORD STO, POI, OUT INI, PRI, HMA HMI, HINQ EPS ?
yes
no
Write error message 51, 52 53
Write error message 51, 54 53
STOP

C.2

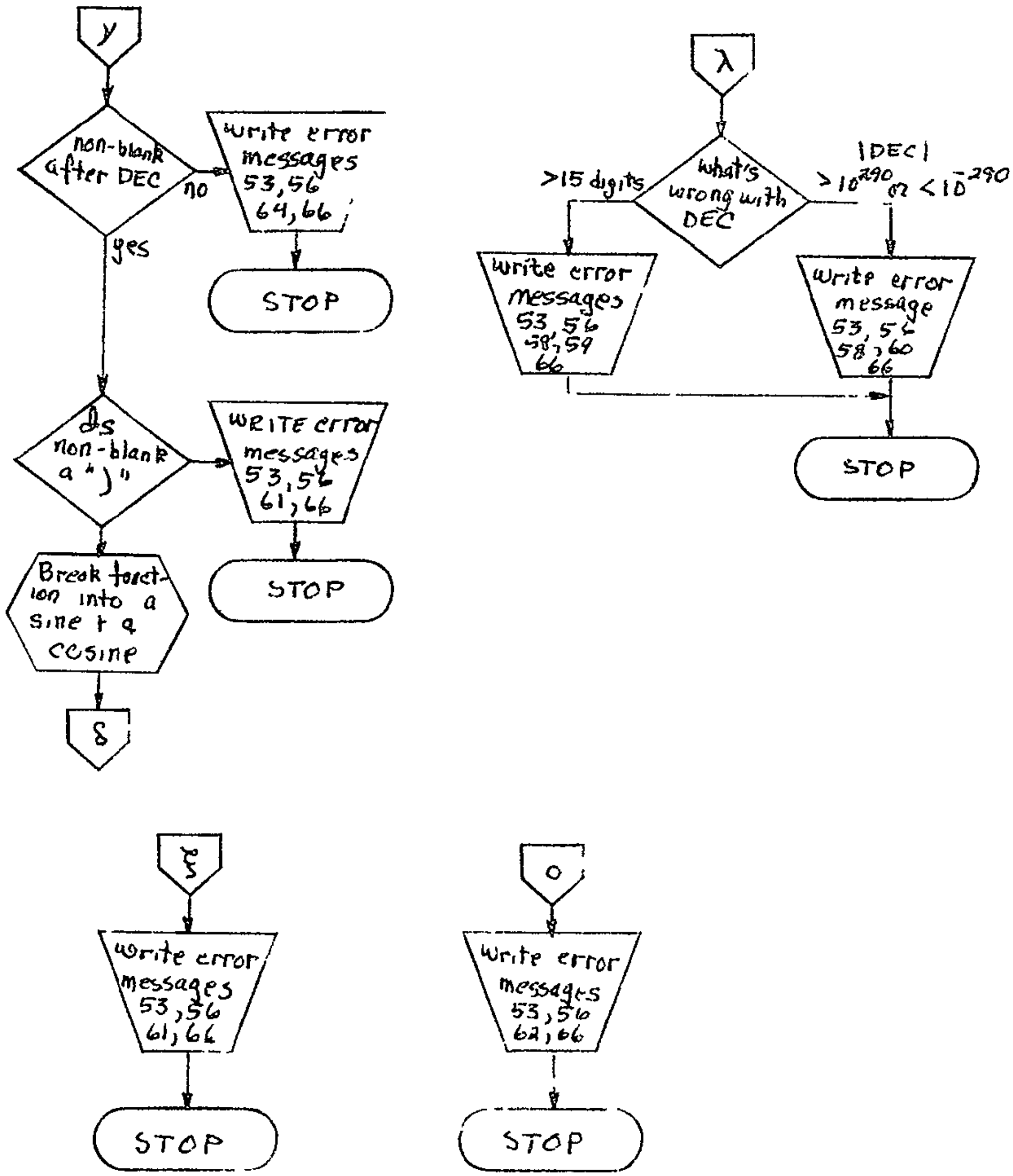
SRCINT


ENTER
Set NINP = 0 NORDER$_i$ = 0 i = 1, NIN
α
Any Card left
no
Create an END card
Read a Card
Is Card the next header card
Were all the sources defined
yes
RETURN
no
Write error messages 53, 56 65, 66
STOP
Is this a continuation of the last
yes
δ
no
Is NINP = 0
yes
no
ISORCE(NINP) = KIN
β
β
KIN = 0 Increase NINP by 1
NINP > 10
yes
Write error messages 57, 66
STOP
no
Find 1st non-blank
Find next blank or equal sign
> 4 letters in source name
yes
Write error messages 53, 56 54, 66
STOP
no
Pack characters between 1st non-blank or = sign in NWORD
Find NWORD on list of source names (Ith source)
Write error messages 53, 56 55, 66
STOP
NORDER(I) = NINP
γ

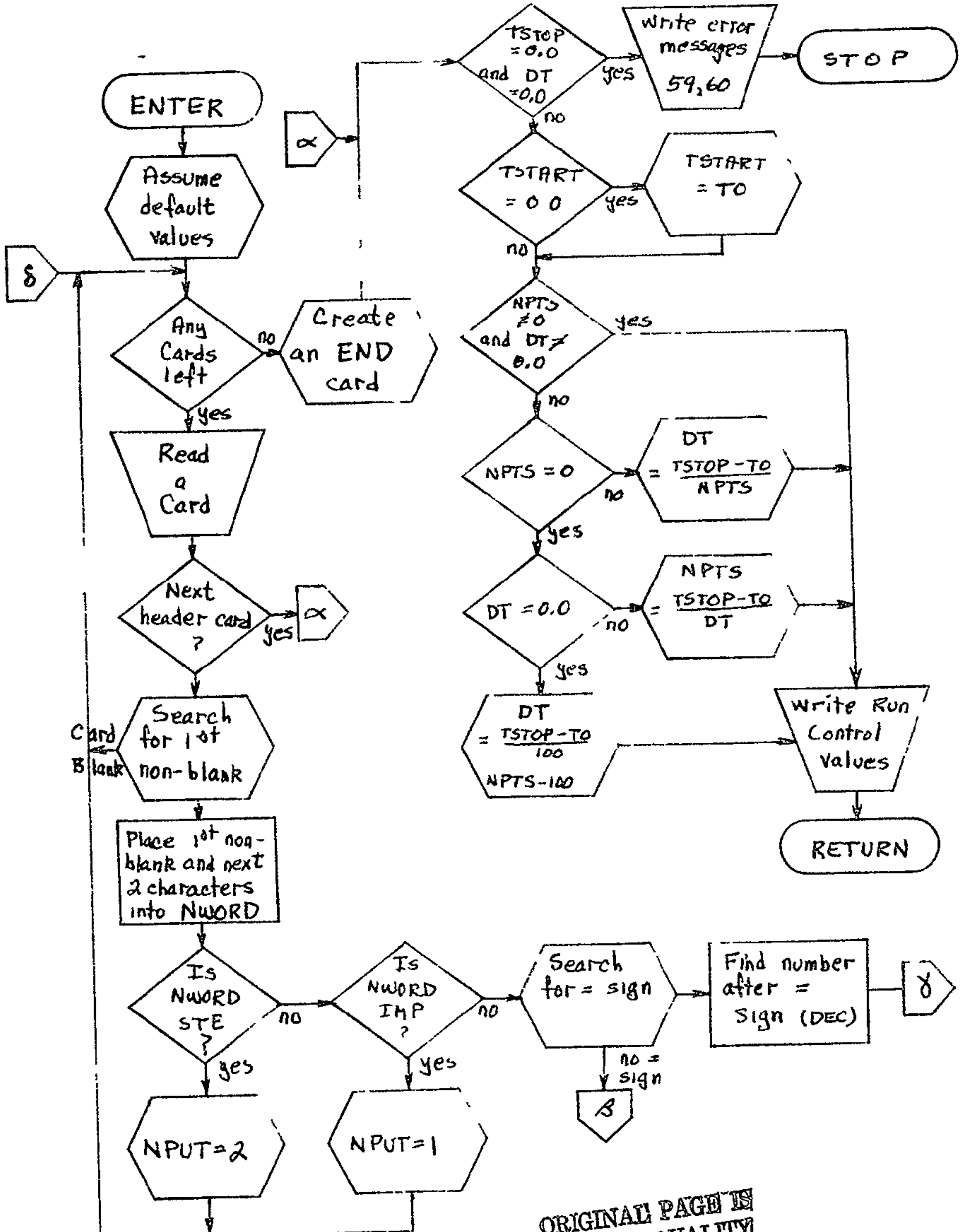

γ
Skip past = sign
no = sign
Write error messages 53, 56 54, 66
STOP
δ
Find next number (DEC)
Error in finding number
λ
DEC = 0.0
no
η
yes
non-blank after DEC
α
no
yes
Is non-blank a + or - sign
δ
yes
no
Skip past the next) or P
ξ
none Present
δ
η
Increase KIN by 1
Is KIN > 20
yes
write error messages 53, 57 66
STOP
Was rest of card blank
yes
Source(NINP, KIN, 1) = DEC NSORCE(NINP, KIN) = 1
α
no
COSAVE = DEC
If non-blank was an * skip to the next non-blank
rest blank
Write error messages 53, 56 61, 66
STOP
Is non-blank a "S"
yes
NSORCE(NINP, KIN) = 2
no
Is non-blank a C
yes
NSORCE(NINP, KIN) = 3
i
no
Is non blank an "I"
no
K
yes
NSORCE(NINP, KIN) = 4 SOURCE(NINP, KIN, 1) = DEC
δ



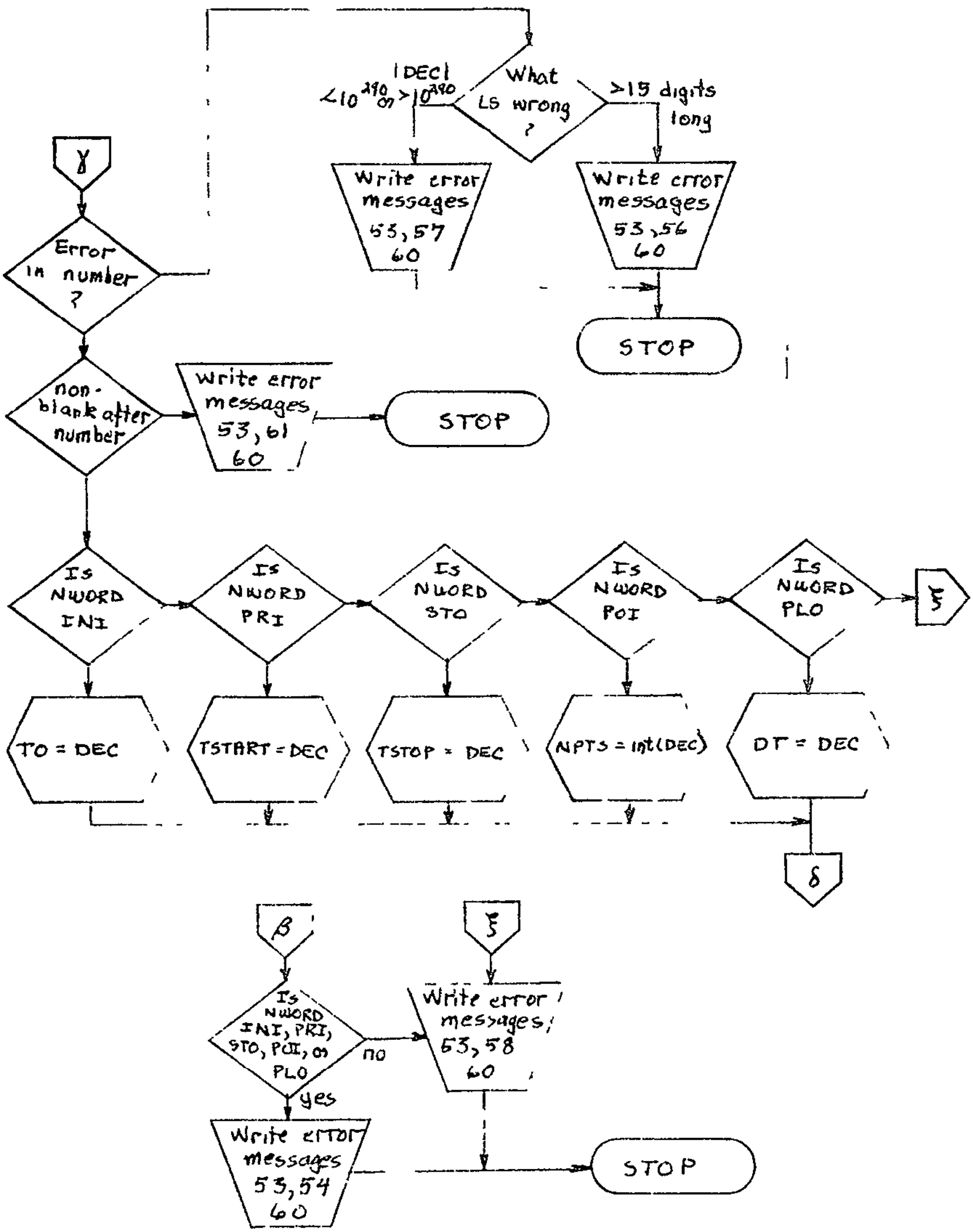
K
Is non blank a + or − sign
no
Write error messages 53, 56 61, 66
STOP
yes
NSORCE(NINP, KIN) = 1
SOURCE(NINP, KIN, 1) = COSAVE
δ
i
Skip over the next 2 characters
Is the next non-blank a "("
no
o
yes
Search for the next number (DEC)
Error in number
yes
λ
μ
μ
If character after number is a *, skip past it
Find next non-blank
Is non-blank a "T"
no
Write error messages 53, 56 63, 66
STOP
yes
SOURCE(NINP, KIN, 2) = DEC
Find the next non blank after T
rest of card blank
Write error messages 53, 56 64, 66
STOP
Is non-blank a ")"
yes
SOURCE(NINP, KIN, 1) = COSAVE
δ
no
Search for the next number (DEC)
Trouble finding number
yes
λ
γ

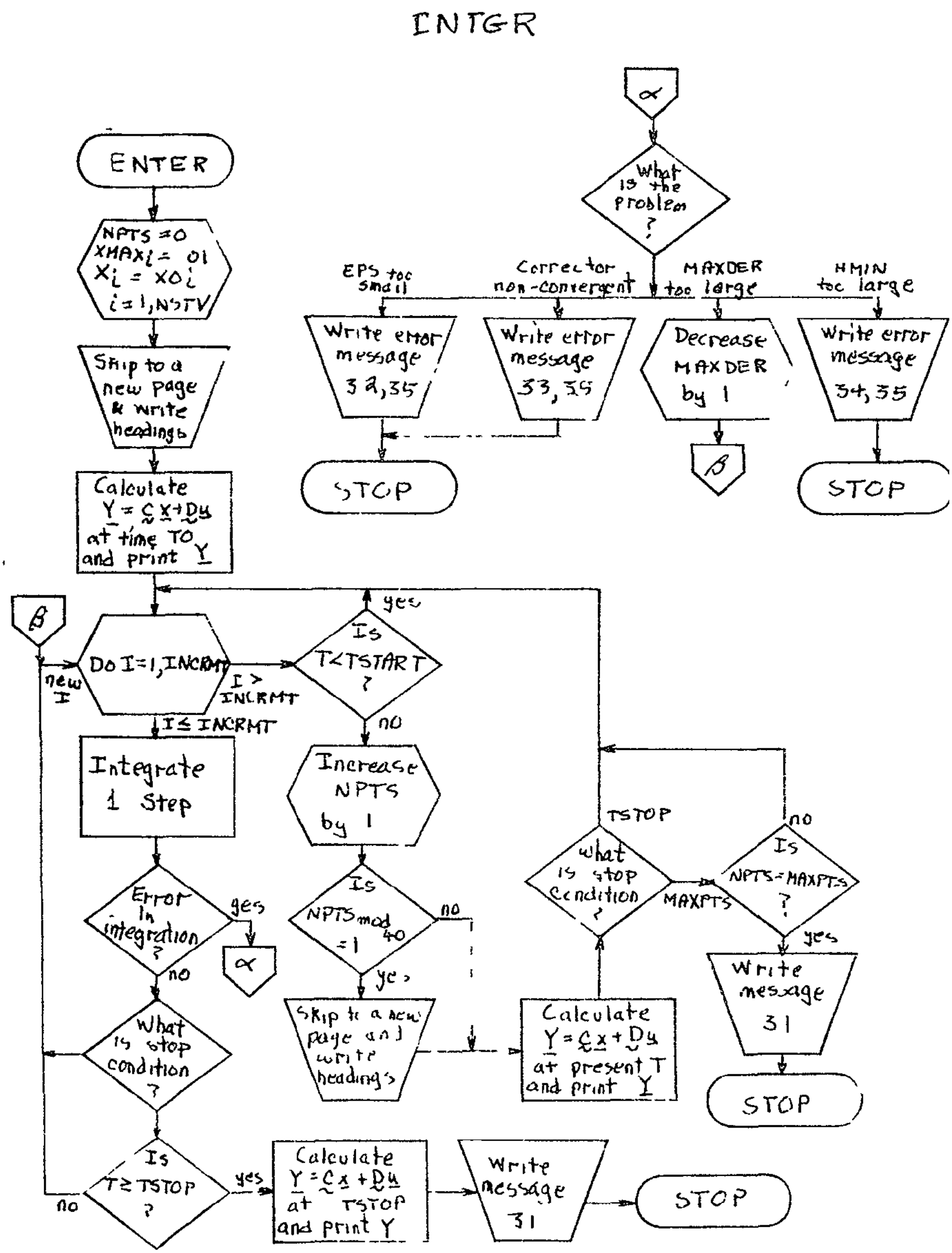

y
non-blank after DEC
no
write error messages 53,56 64,66
STOP
yes
Is non-blank a ")"
WRITE error messages 53,56 61,66
STOP
Break function into a sine + a cosine
δ
λ
what's wrong with DEC
>15 digits
|DEC| > 10^290 or < 10^-290
write error messages 53 56 58,59 66
write error message 53, 56 58, 60 66
STOP
ξ
write error messages 53,56 61,66
STOP
o
write error messages 53,56 62,66
STOP

RUNCRL

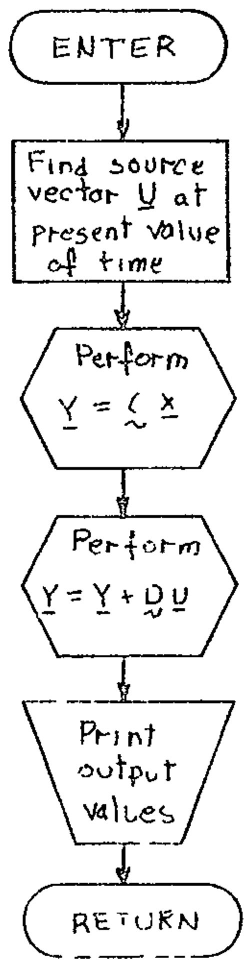


ORIGINAL PAGE IS OF POOR QUALITY

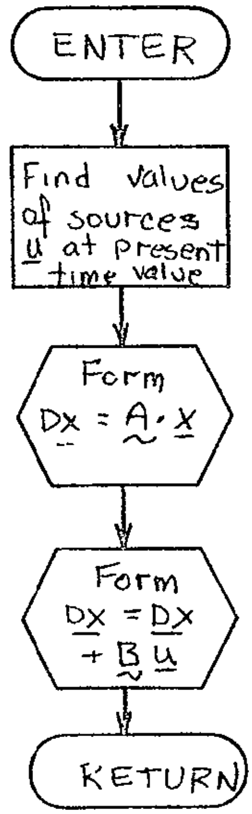

What is wrong ?
|DEC| < 10^-290 or > 10^290
>15 digits long
Write error messages 53, 57 60
Write error messages 53, 56 60
STOP
γ
Error in number ?
non-blank after number
Write error messages 53, 61 60
STOP
Is NWORD INI
Is NWORD PRI
Is NWORD STO
Is NWORD POI
Is NWORD PLO
ξ
TO = DEC
TSTART = DEC
TSTOP = DEC
NPTS = int(DEC)
DT = DEC
δ
β
ξ
Is NWORD INI, PRI, STO, POI, or PLO
no
yes
Write error messages 53, 58 60
Write error messages 53, 54 60
STOP


INTGR
α
What is the problem ?
EPS too small
Corrector non-convergent
MAXDER too large
HMIN too large
Write error message 32,35
Write error message 33,35
Decrease MAXDER by 1
Write error message 34,35
STOP
β
STOP
ENTER
NPTS = 0
XMAXi = 0I
Xi = XOi
i = 1,NSTV
Skip to a new page & write headings
Calculate Y = Cx + Du at time TO and print Y
β
Do I=1,INCRMT
new I
I > INCRMT
Is T<TSTART ?
yes
no
I ≤ INCRMT
Integrate 1 Step
Increase NPTS by 1
Error in integration ?
yes
α
no
Is NPTS mod 40 = 1
no
yes
What is stop condition ?
Skip to a new page and write headings
Calculate Y = Cx + Du at present T and print Y
what is stop condition ?
TSTOP
MAXPTS
Is NPTS=MAXPTS ?
no
yes
Write message 31
STOP
Is T≥TSTOP ?
no
yes
Calculate Y = Cx + Du at TSTOP and print Y
Write message 31
STOP

DATPNT

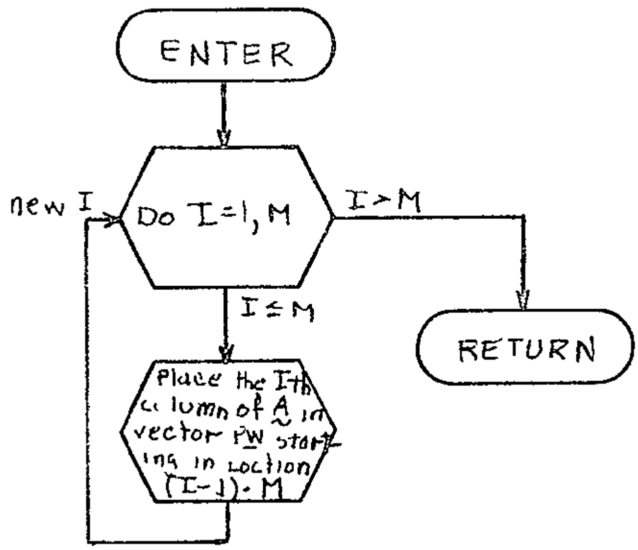

ENTER
Find source vector U at present value of time
Perform
Y = C x
Perform
Y = Y + D U
Print output values
RETURN

DIFFUN

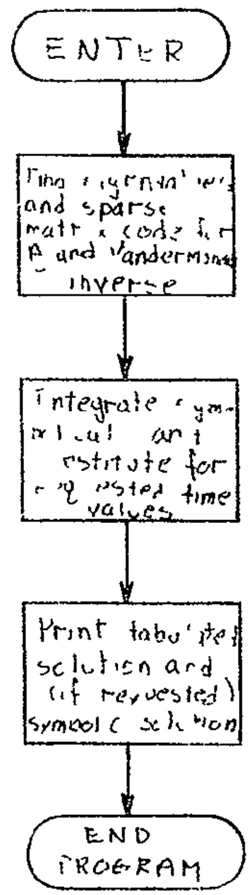


ORIGINAL PAGE IS
OF POOR QUALITY

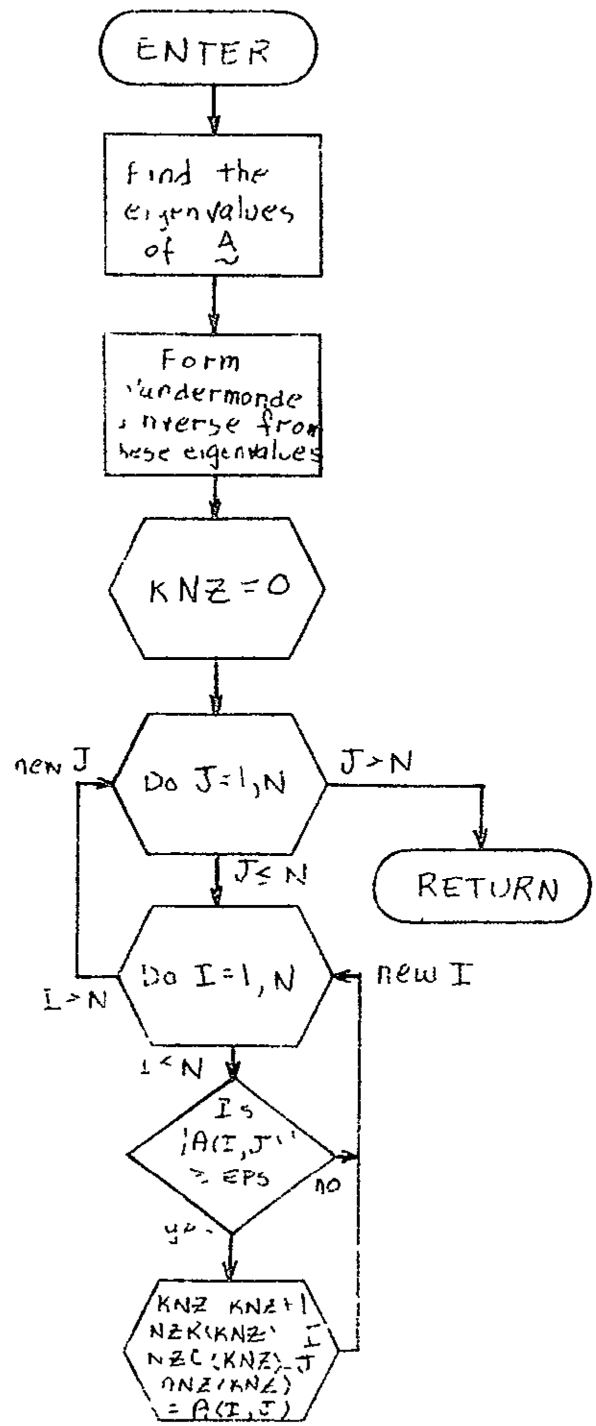
PEDERV


ENTER
new I
Do I=1, M
I > M
I ≤ M
RETURN
Place the Ith column of A in vector PW starting in loction (I-1)·M

MATRIX

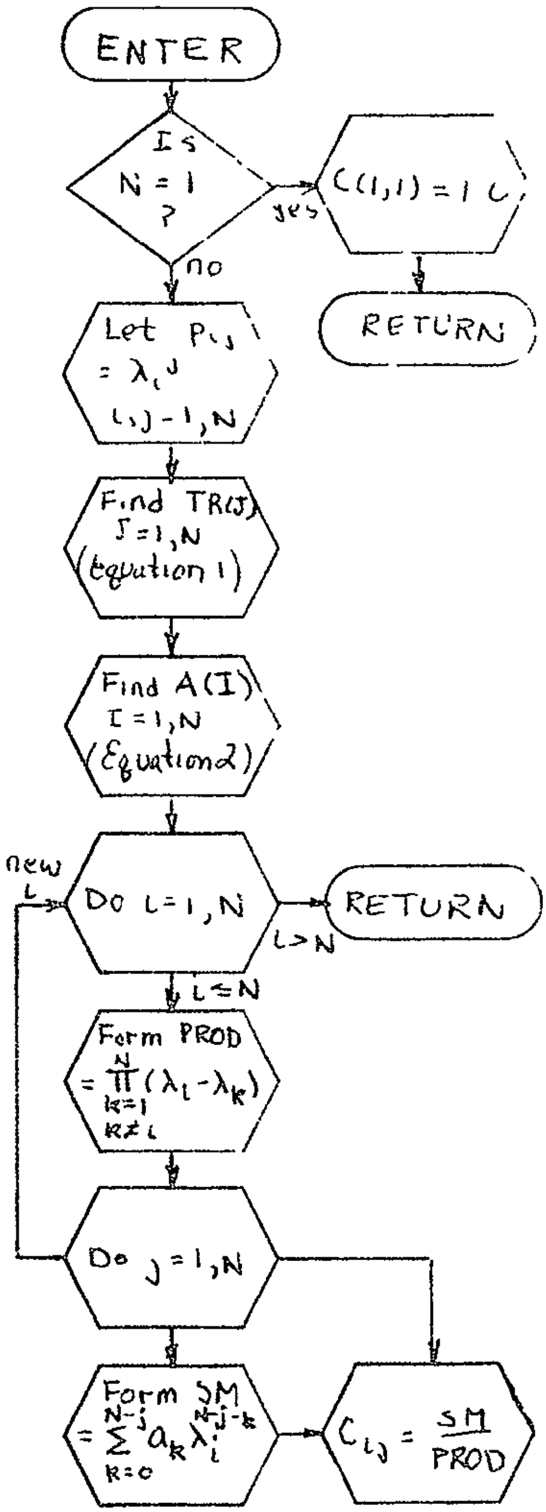

ENTER
and sparse
inverse
Integrate
for
time
values
Print
solution and
(if requested)
END
PROGRAM

SETSVE



ENTER
Find the eigenvalues of A
Form Vandermonde inverse from these eigenvalues
KNZ = 0
Do J = 1, N
new J
J > N
RETURN
J ≤ N
Do I = 1, N
new I
I > N
I ≤ N
Is |A(I, J)| ≥ EPS
no
yes
KNZ = KNZ + 1
NZR(KNZ) = I
NZC(KNZ) = J
ANZ(KNZ) = A(I, J)

VANINV



Equation 1

$$TR(J) = Tr\,\underset{\sim}{A}^J = \sum_{l=1}^{N} \lambda_l^J = \sum_{l=1}^{N} P_{J,l}$$

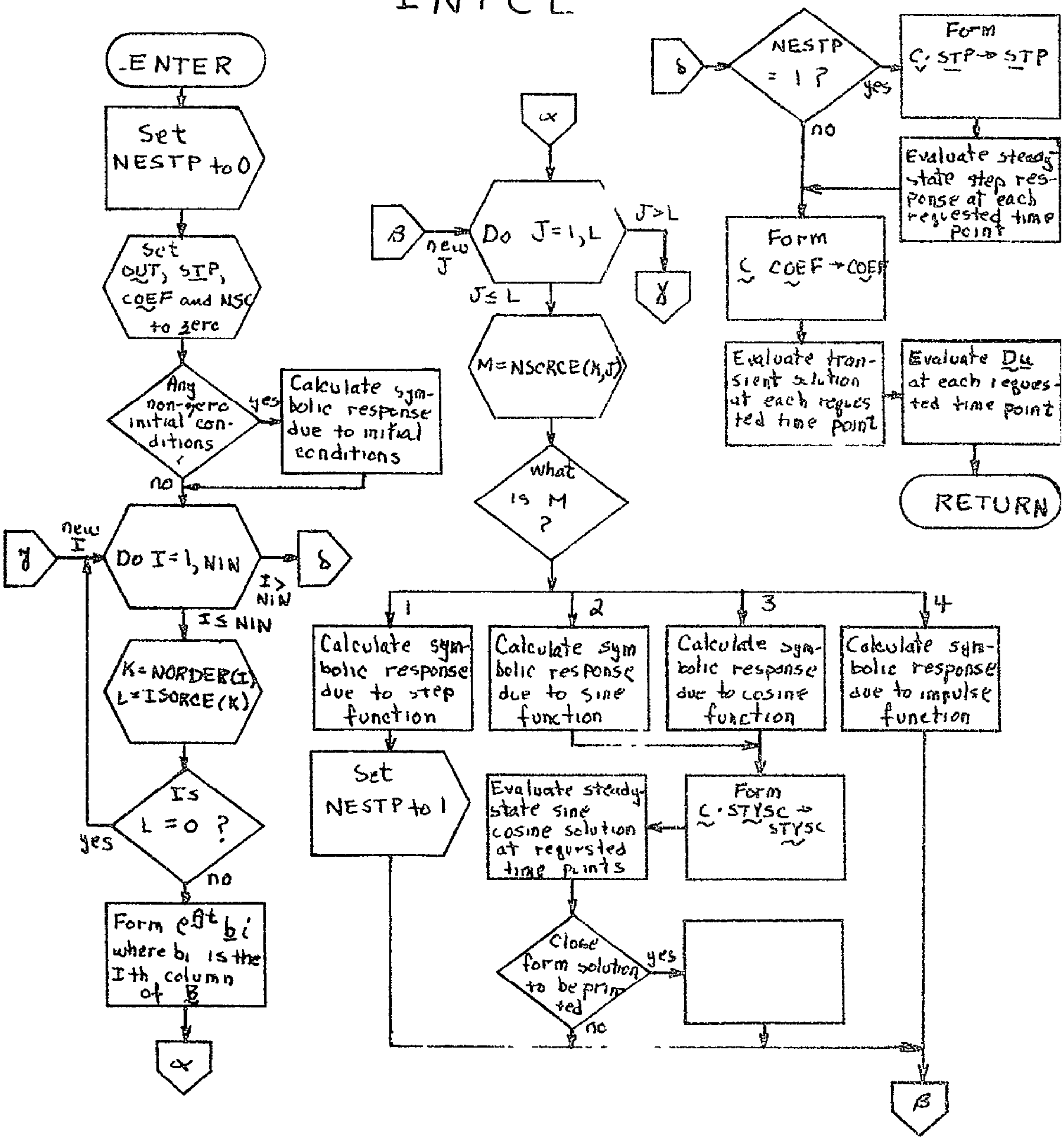
Equation 2

$$|\underset{\sim}{A} - \lambda \underset{\sim}{I}| = \lambda^n + a_1 \lambda^{n-1} + \cdots + a_n$$

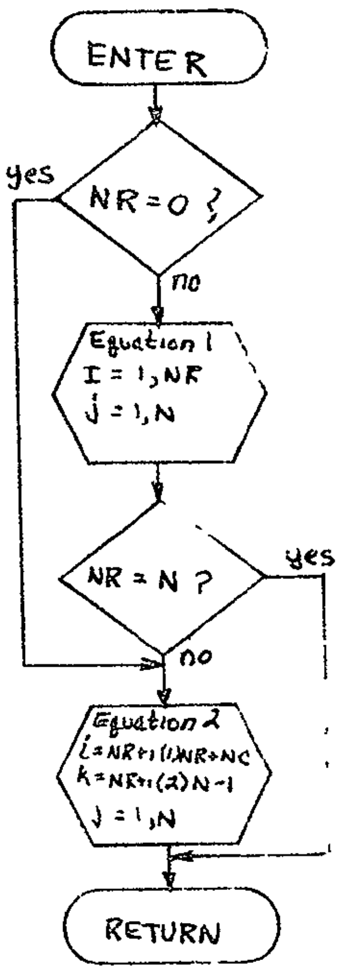
$$a_0 = 1$$

$$A(I) = a_i = -\frac{1}{i} \sum_{m=1}^{i} a_{i-m}\, TR(m)$$

INTCL


ENTER
Set NESTP to 0
Set OUT, STP, COEF and NSC to zero
Any non-zero initial conditions
yes
Calculate symbolic response due to initial conditions
no
new I
Do I=1,NIN
I > NIN
I ≤ NIN
K=NORDER(I) L=ISORCE(K)
Is L = 0 ?
yes
no
Form e^{At} b_i where b_i is the Ith column of B
β
new J
Do J=1,L
J>L
J≤L
M=NSCRCE(K,J)
What is M ?
1
2
3
4
Calculate symbolic response due to step function
Calculate symbolic response due to sine function
Calculate symbolic response due to cosine function
Calculate symbolic response due to impulse function
Set NESTP to 1
Evaluate steady-state sine cosine solution at requested time points
Form C·STYSC → STYSC
Close form solution to be printed
yes
no
NESTP = 1 ?
yes
no
Form C·STP → STP
Evaluate steady-state step response at each requested time point
Form C COEF → COEF
Evaluate transient solution at each requested time point
Evaluate Du at each requested time point
RETURN

SVECOS



Equation 1

$$B_i = ROOT_i^2 + OMEGA^2$$

$$C_i = AMPL\, e^{-ROOT_i\, TO}\left[-ROOT_i \cdot \cos(OMEGA\ TO) + OMEGA \sin(OMEGA\ TO)\right]/B$$

$$D_i = -AMPL \cdot ROOT_i / B_i$$

$$E_i = AMPL \cdot OMEGA / B_i$$

$$STYCOS_{j,1} = STYCOS_{j1} + P_{ji} \cdot E_i$$

$$STYCOS_{j,2} = STYCOS_{j,2} + P_{ji}\ D_i$$

$$CFCOS_{ji} - CFCOS_{ji} - P_{ji}\ C_i$$

Equation 2

$$ALFA_K = Re[ROOT_K] \quad , \quad BETA_K = Im[ROOT_K]$$

$$\omega_{1,K} = BETA_K - OMEGA \quad , \quad \omega_{2,K} = BETA_K + OMEGA$$

$$C_{1,K} = ALFA_K^2 + \omega_{1,K}^2 \quad , \quad C_{2,K} = ALFA_K^2 + \omega_{2,K}^2$$

$$T_{1,K} = -e^{-ALFA_K\ TO}\left[\omega_{1,K} \sin(\omega_{1,K} TO) - ALFA_K \cos(\omega_{1,K} \cdot TO)\right]/C1$$

$$\quad - e^{-ALFA_K\ TO}\left[\omega_{2,K} \sin(\omega_{2,K}\ TO) - ALFA_K \cos(\omega_{2,K} \cdot TO)\right]/C2$$

$$T_{2,K} = -e^{-ALFA_K \cdot TO}\left[-ALFA_K \sin(\omega_{1,K} \cdot TO) - \omega_{1,K} \cos(\omega_{1,K}\ TO)\right]/C1$$

$$\quad - e^{-ALFA_K\ TO}\left[-ALFA_K \sin(\omega_{2,K} \cdot TO) - \omega_{2,K} \cos(\omega_{2,K} \cdot TO)\right]/C2$$

$$PR_{ji} = AMPL\ Re[P_{ji}]$$

$$PI_{ji} = AMPL \cdot Im[P_{ji}]$$

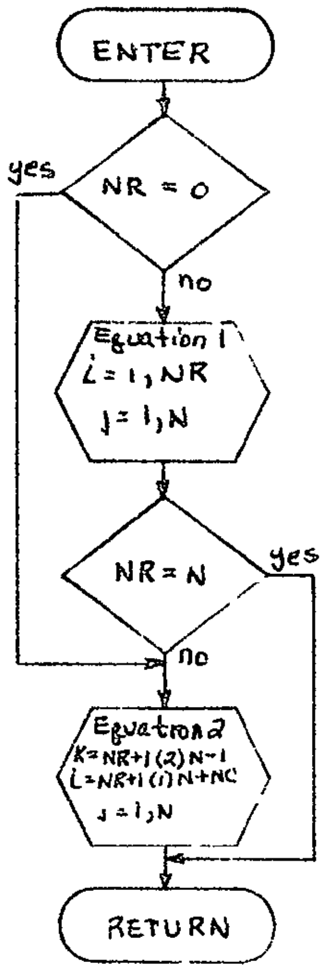
$$STYCOS_{j1} = STYCOS_{j1} + (-PR_{ji}\ \omega_{1,K} + PI_{ji} \cdot ALFA_K)/C_{1,K} + (PI_{ji} \cdot ALFA_K + PR_{ji}\ \omega_{2,K})/C_{2,K}$$

$$STYCOS_{j2} = STYCOS_{j2} + (-PR_{ji} \cdot ALFA_K - PI_{ji}\ \omega_{1,K})/C_{1,K} + (-PR_{ji}\ ALFA_K - PI_{ji}\ \omega_{2,K})/C_{2,K}$$

$$CFCOS_{j,K} = CFCOS_{j,K} - T_{1,K}\ PI_{ji} + T_{2,K}\ PR_{ji}$$

$$CFCOS_{j,K+1} = CFCOS_{j,K+1} + T_{1,K} PR_{ji} + T_{2,K}\ PI_{ji}$$

SVESIN



Equation 1

$B_i = ROOT_i^2 + OMEGA^2$

$C_i = AMPL\ e^{-ROOTi\ TO}\left[-ROOT_i\ \sin(OMEGA\cdot TO) - OMEGA\cdot \cos(OMEGA\cdot TO)\right]/B_i$

$D_i = AMPL\cdot(-ROOT_i/B_i)\ ,\ E_i = AMPL\ (-OMEGA/B_i)$

$STYSIN_{j1} = STYSIN_{j,1} + P_{ji}\cdot D_i$

$STYSIN_{j,2} = STYSIN_{j,2} + P_{ji}\ E_i$

$CFSIN_{ji} = CFSIN_{ji} - P_{ji}\ C_i$

Equation 2

$ALFA_K = Re\left[ROOT(K)\right]\ ,\ BETA_K = Im\left[ROOT_K\right]$

$\omega_{1,K} = BETA_K - OMEGA\ ,\ \omega_{2,K} = BETA_K + OMEGA$

$C1_{,K} = ALFA_K^2 + \omega_{1,K}^2\ ,\ C_{2,K} = ALFA_K^2 + \omega_{2,K}^2$

$K_{1,K} = -e^{-ALFA_K\ TO}\left[\omega_{1,K}\cdot\cos(\omega_{1,K}\cdot TO) + ALFA_K\ \sin(\omega_{1,K}\cdot TO)\right]/C1$
$\quad + e^{-ALFA_K\ TO}\left[\omega_{2,K}\ \cos(\omega_{2,K}\cdot TO) + ALFA_K\ \sin(\omega_{2,K}\cdot TO)\right]/C2$

$K_{2,K} = -e^{-ALFA_K\cdot TO}\left[\omega_{1,K}\cdot\sin(\omega_{1,K}\ TO) - ALFA_K\cos(\omega_{1,K}\cdot TO)\right]/C1$
$\quad + e^{-ALFA_K\ TO}\left[\omega_{2,K}\ \sin(\omega_{2,K}\ TO) - ALFA_K\cos(\omega_{2,K}\cdot TO)\right]/C2$

$PR_{ji} = AMPL\cdot Re\left[P_{ji}\right]$

$PI_{ji} = AMPL\ Im\left[P_{ji}\right]$

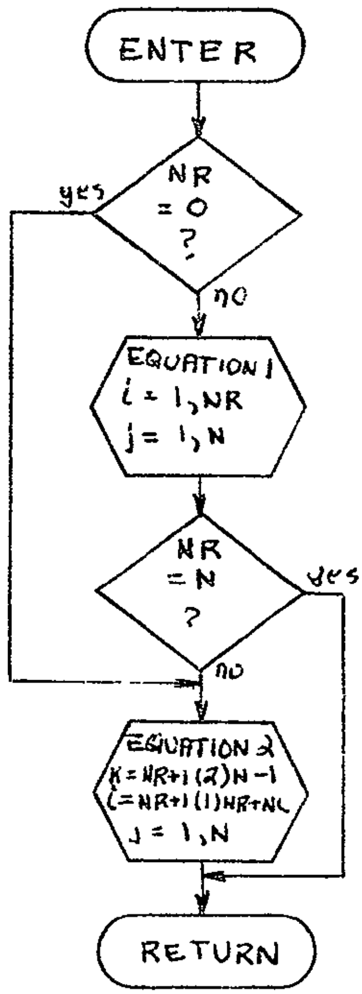
$STYSIN_{j,2} = STYSIN_{j2} + (PR_{ji}\cdot\omega_{1,K} - PI_{ji}\ ALFA_K)/C1 + (PI_{ji}\ ALFA_K - PR_{ji}\ \omega_{2,K})/C2$

$STYSIN_{j,1} = STYSIN_{j,1} + (-PR_{ji}\ ALFA_K - PI_{ji}\ \omega_{1,K})/C1 + (-PR_{ji}\cdot ALFA_K - PI_{ji}\cdot\omega_{2,K})/C2$

$CFSIN_{j,k+1} = CFSIN_{j,k+1} + K_{1,K}\ PR_{ji} + K_{2,K}\ PI_{ji}$

$CFSIN_{j,k} = CFSIN_{j,k} - K_{1,K}\cdot PI_{ji} + K_{2,K}\ PR_{ji}$

SVESTP



Equation 1

$$A_L = -\frac{AMPL}{ROOT_L}$$

$$B_L = \frac{AMPL\ e^{-ROOT_L\,TO}}{ROOT_L}$$

$$COEF_{JL} = COEF_{JL} + P_{JL}\ B_L$$

$$STP_J = STP_J + P_{JL} \cdot A_L$$

Equation 2

$$\alpha_K = Re[ROOT_K]$$

$$\beta_K = Im[ROOT_K]$$

$$C_K = \frac{2\,AMPL}{\alpha_K^2 + \beta_K^2}$$

$$A_K = C_K\ e^{-\alpha_K TO}[\alpha_K \cos(\beta_K TO) - \beta_K \sin(\beta_K TO)$$

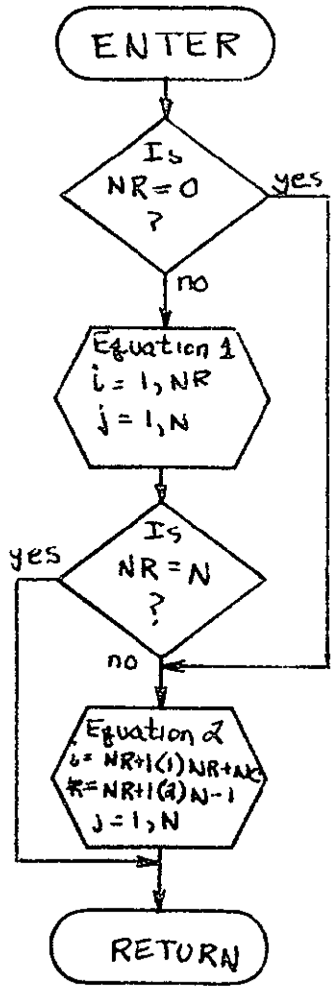
$$B_K = C_K\ e^{-\alpha_K TO}[\beta_K \cos(\beta_K TO) - \alpha_K \sin(\beta_K TO$$

$$COEF_{J,K} = COEF_{J,K} - Im[P_{JL}]A_K + Re[P_{JL}]B_K$$

$$COEF_{J,K+1} = COEF_{J,K+1} + Re[P_{JL}]A_K + Im[P_{JL}]B_K$$

$$STP_J = STP_J + C_K[-\alpha_K Re(P_{JL}) - \beta_K Im(P_{JL})]$$

INIVAL



Equation 1

$$CF_i = AMPL\ e^{-ROOT_i \cdot TO}$$

$$COEFO_{ji} = COEFO_{ji} + P_{ji}\ CF_i$$

Equation 2

$$\alpha_k = Re[ROOT_k]$$

$$\beta_k = Im[ROOT_k]$$

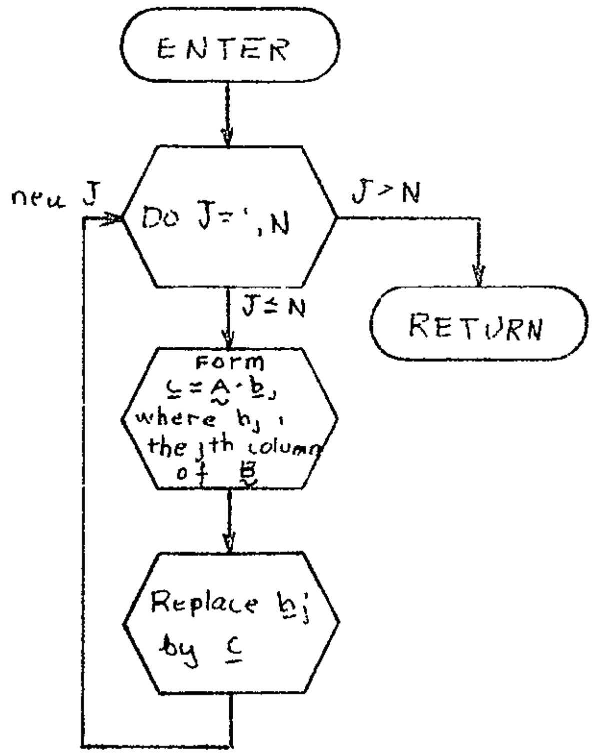
$$A_k = AMPL \cdot Re[P_{ji}]$$

$$B_k = AMPL \cdot Im[P_{ji}]$$

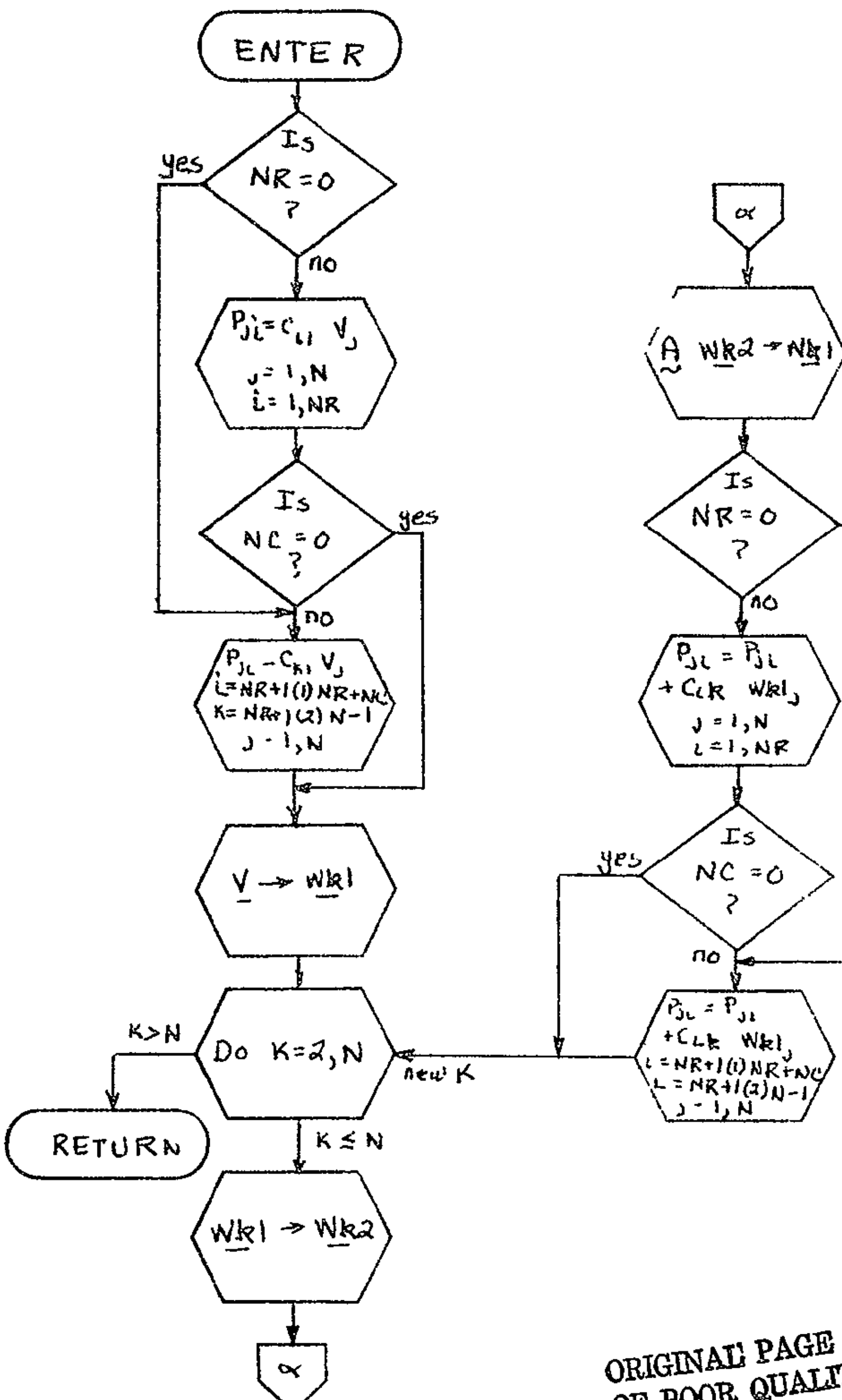
$$COEFO_{j,k} = COEFO_{j,k} + 2 \cdot e^{-\alpha_k TO}[A_k \cos(\beta_k TO) + B_k \sin(\beta_k \cdot TO)]$$

$$COEFO_{j,k+1} = COEF_{j,k+1} + 2\ e^{-\alpha_k TO}[A_k \sin(\beta_k TO) + B_k \cos(\beta_k TO)]$$

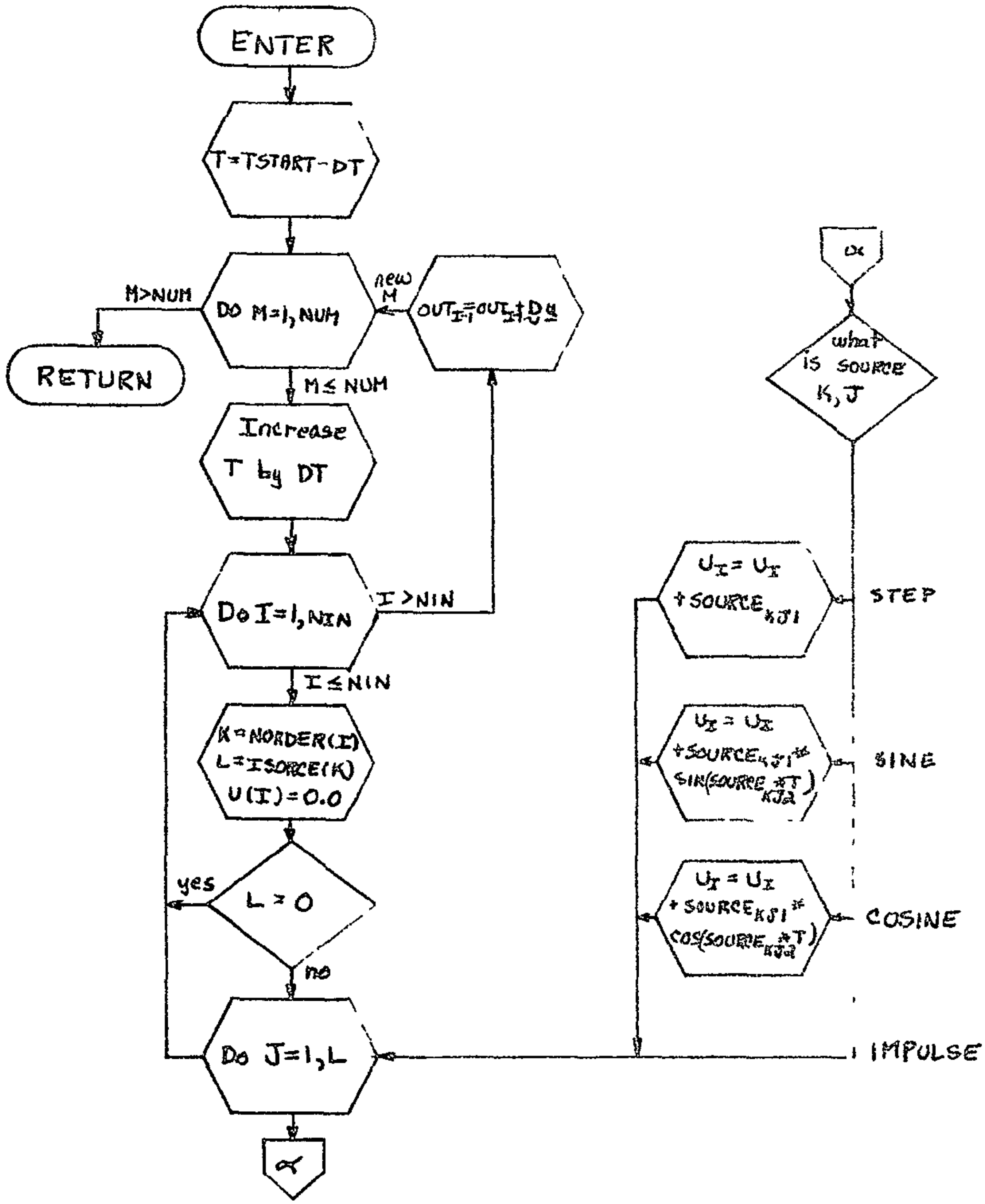
MPROD


ENTER
new J
Do J=1,N
J>N
RETURN
J≤N
Form
c = A · b_j
where b_j i
the jth column
of B
Replace b_j
by c

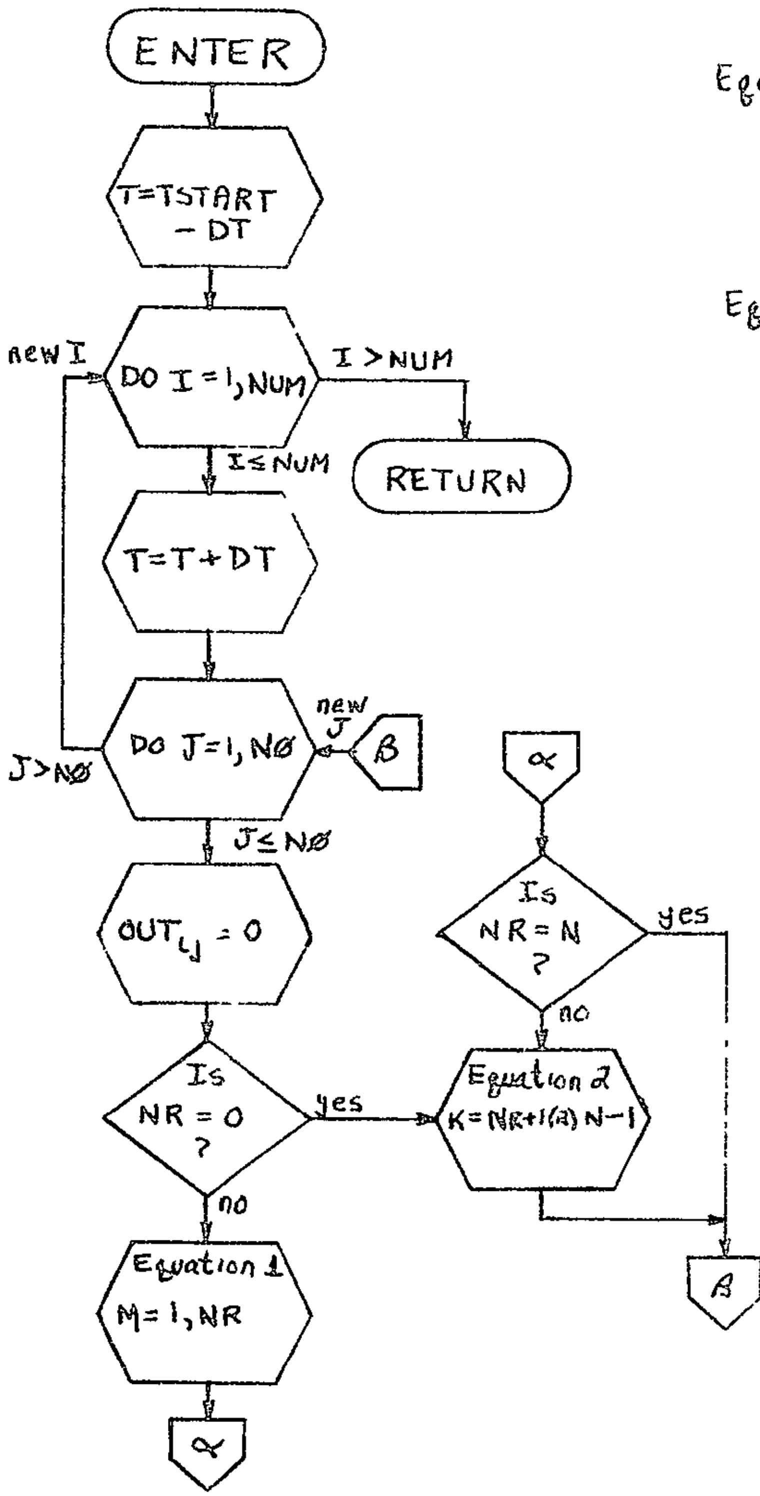
EAT


ENTER
Is NR = 0 ?
yes
no
Is NC = 0 ?
yes
no
V → Wk1
Do K=2,N
K>N
RETURN
K ≤ N
Wk1 → Wk2
new K
α
A Wk2 → Wk1
Is NR = 0 ?
yes
no
Is NC = 0 ?
yes
no
α

SCEVAL


ENTER
T=TSTART-DT
DO M=1,NUM
M>NUM
RETURN
M≤NUM
new M
Increase T by DT
Do I=1,NIN
I>NIN
I≤NIN
K=NORDER(I)
L=ISORCE(K)
U(I)=0.0
L=0
yes
no
Do J=1,L
α
what is SOURCE K,J
STEP
SINE
COSINE
IMPULSE

TRANS



Equation 1

$$OUT_{ij} = OUT_{ij} + COEF_{jM}\, e^{ROOT_M T}$$

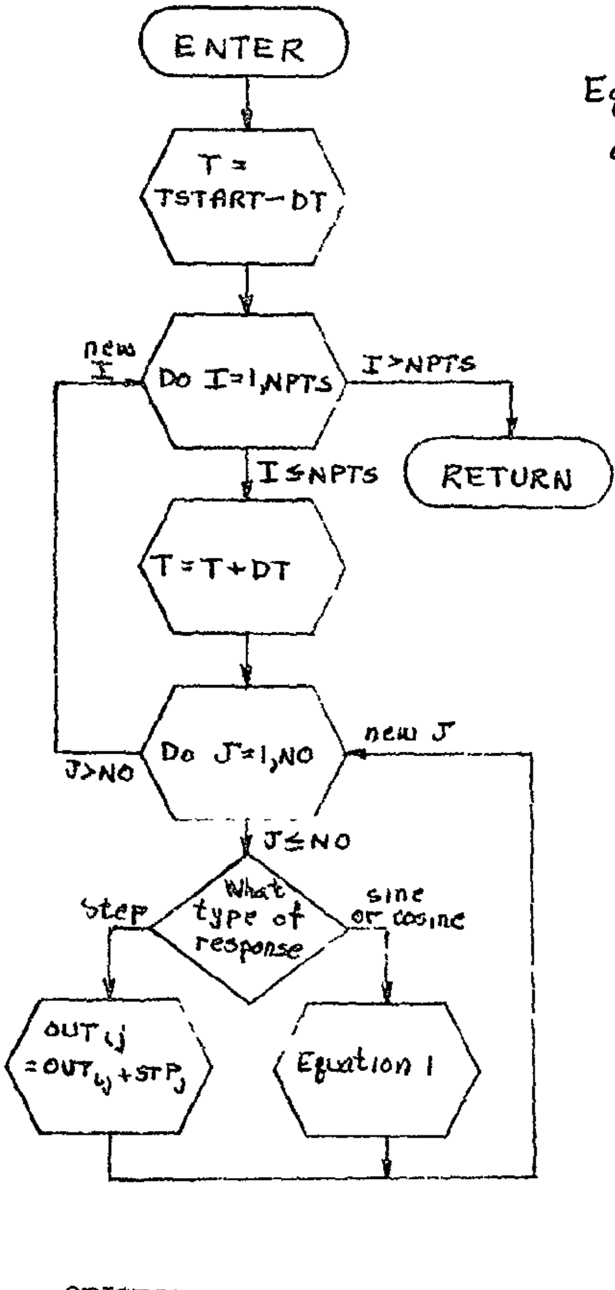
Equation 2

$$\alpha_k = Re\,[ROOT_k]$$

$$\beta_k = Im\,[ROOT_k]$$

$$OUT_{ij} = OUT_{ij} + [\,COEF_{jk} \sin(\beta_k T) + COEF_{j,k+1} \cos(\beta_k T)\,]\, e^{\alpha_k T}$$

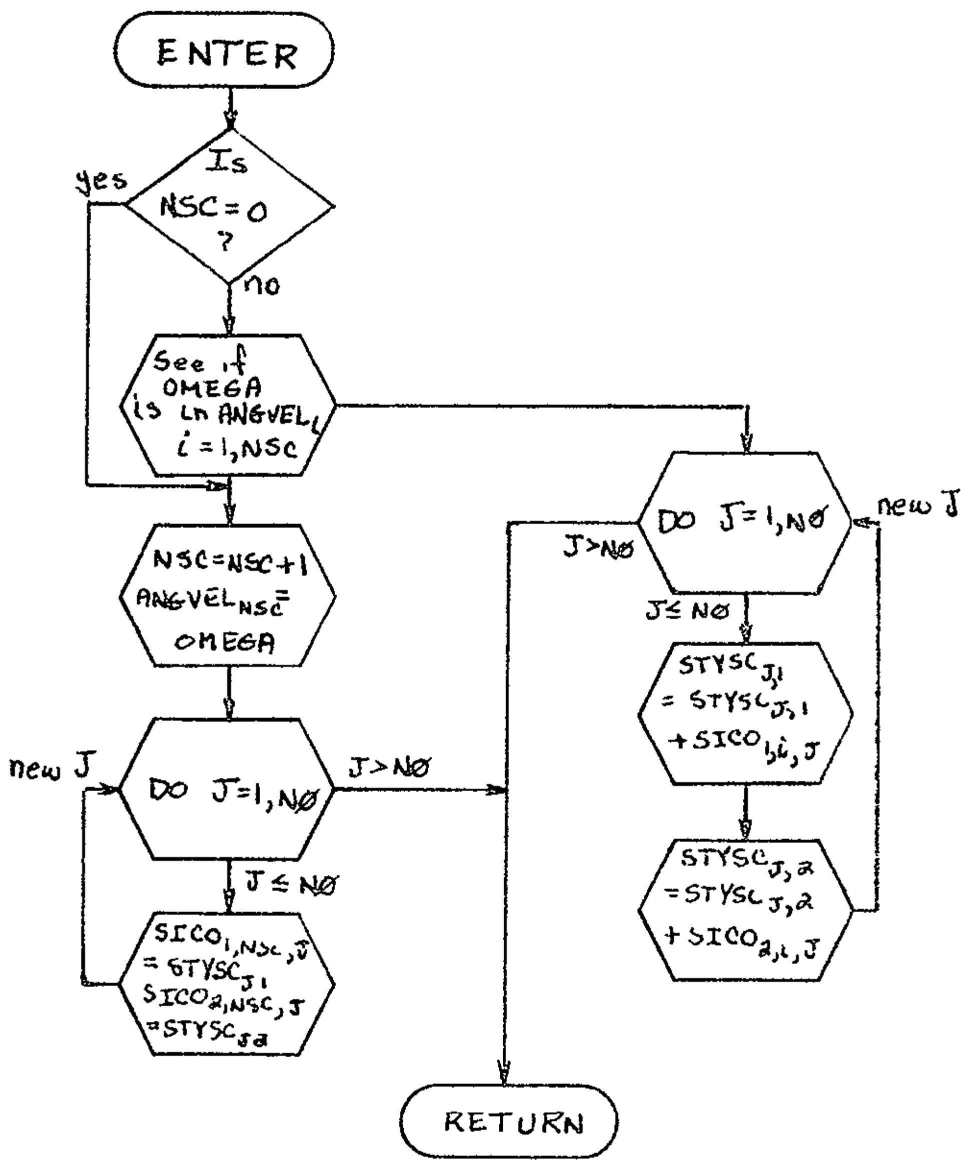
STEADY



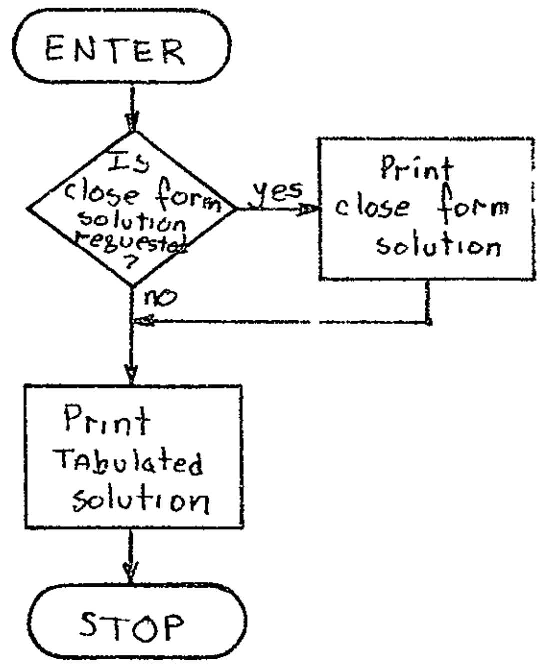
Equation 1

$$OUT_{ij} = OUT_{ij} + STYSC_{j1} \cdot \sin(OMEGA \cdot T) + STYSC_{j2} \cos(OMEGA \cdot T)$$

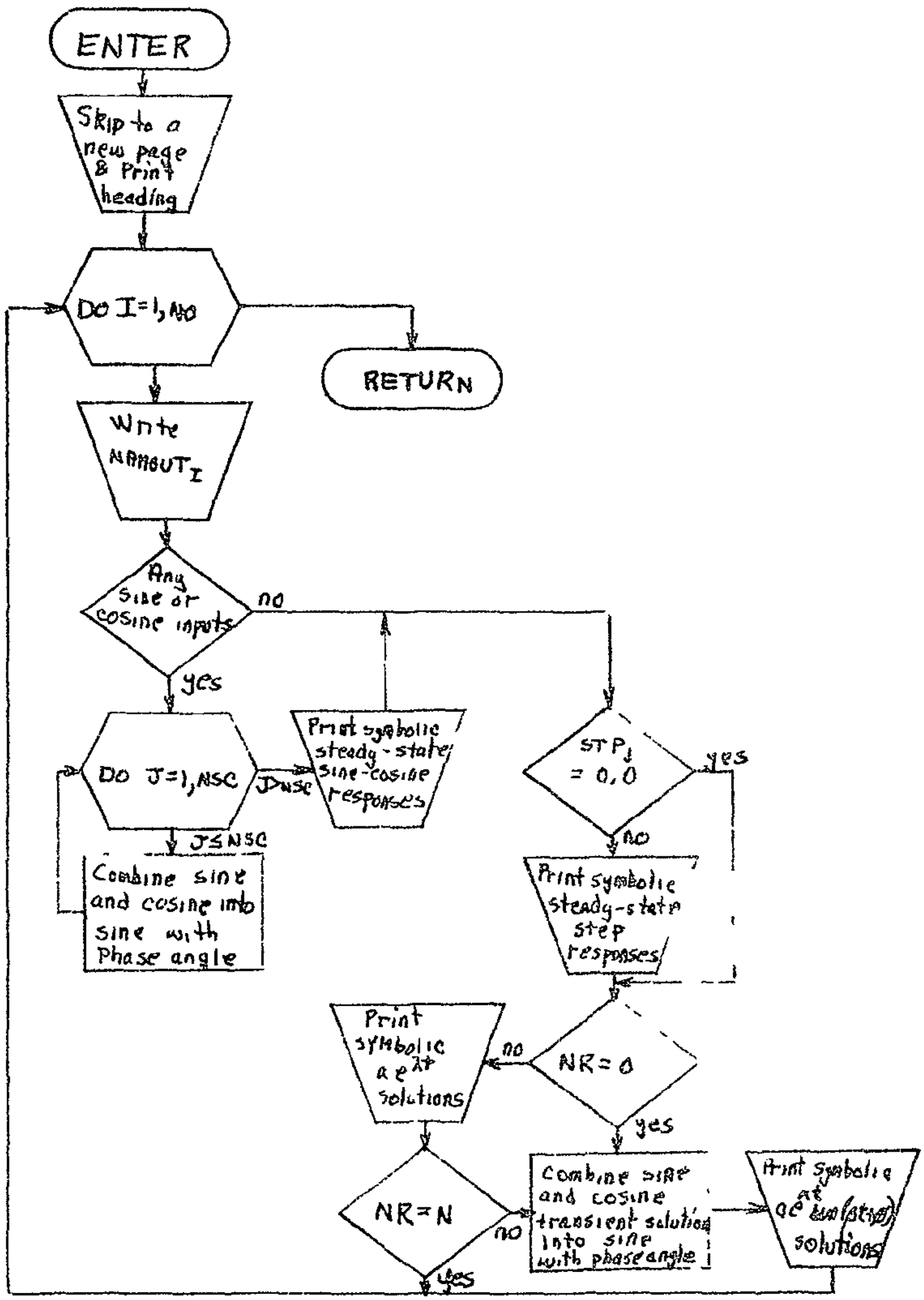
STORE


ENTER
Is NSC = 0 ?
yes
no
See if OMEGA is in ANGVEL_i i = 1,NSC
NSC=NSC+1
ANGVEL_NSC = OMEGA
new J
DO J=1,NØ
J>NØ
J ≤ NØ
SICO_1,NSC,J = STYSC_J,1
SICO_2,NSC,J = STYSC_J,2
DO J=1,NØ
new J
J>NØ
J≤ NØ
STYSC_J,1 = STYSC_J,1 + SICO_1,i,J
STYSC_J,2 = STYSC_J,2 + SICO_2,i,J
RETURN

OUT PNT

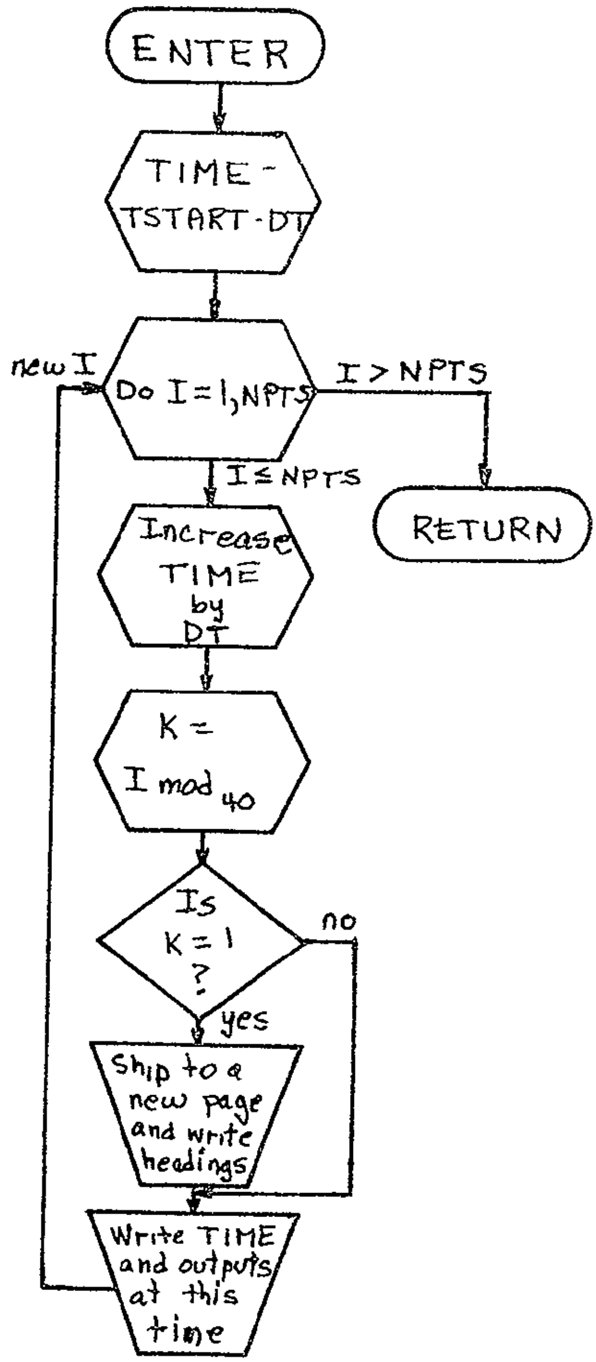

ENTER
Is close form solution requested ?
yes
no
Print close form solution
Print Tabulated solution
STOP

CLSOL

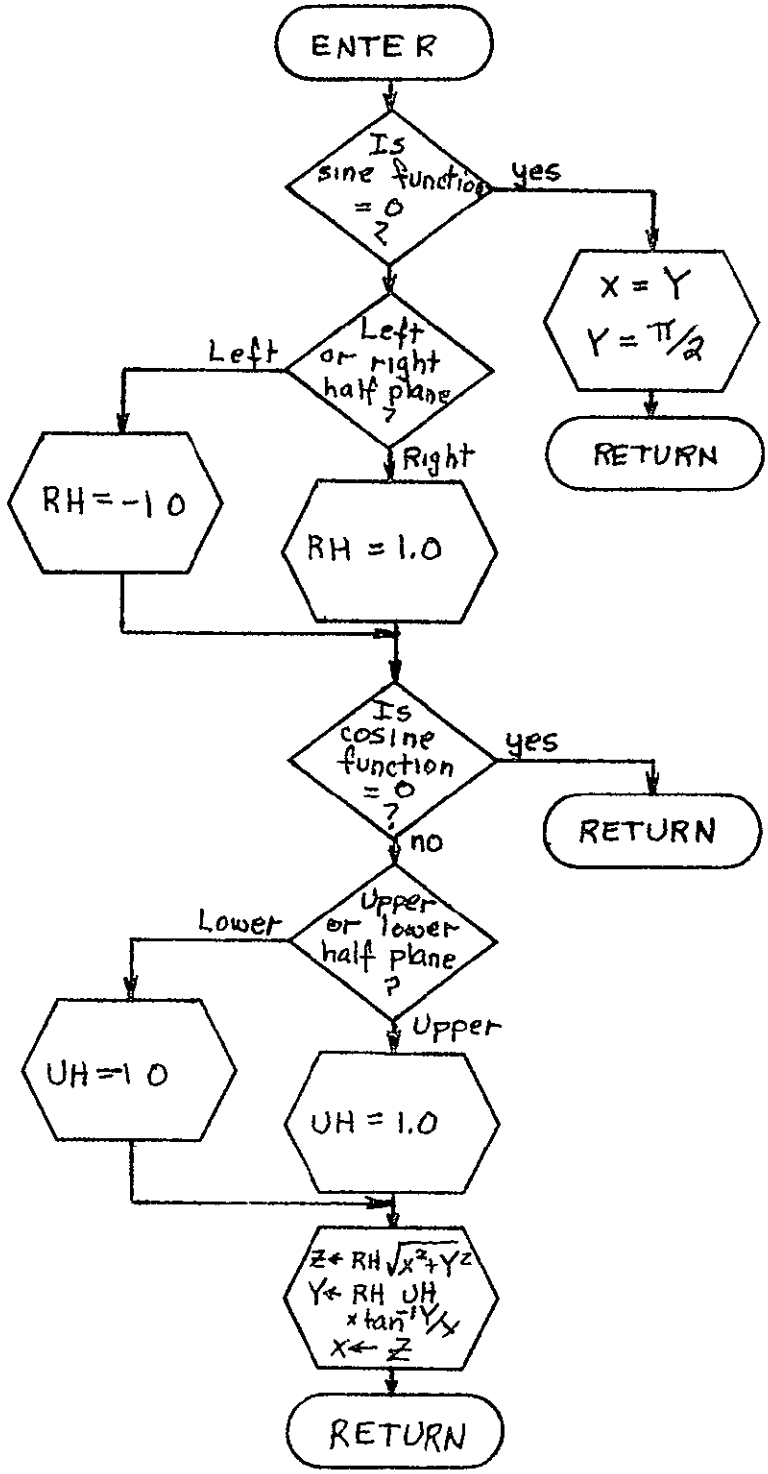


ENTER
Skip to a new page & Print heading
DO I=1,NO
RETURN
Write NAMOUT$_I$
Any sine or cosine inputs
no
yes
DO J=1,NSC
J>NSC
J≤NSC
Combine sine and cosine into sine with Phase angle
Print symbolic steady-state sine-cosine responses
STP$_J$ = 0.0
yes
no
Print symbolic steady-state step responses
NR=0
no
yes
Print symbolic a e$^{\lambda t}$ solutions
NR=N
no
yes
Combine sine and cosine transient solution into sine with phase angle
Print symbolic solutions

TABPNT


ENTER
TIME = TSTART - DT
Do I = 1, NPTS
new I
I > NPTS
I ≤ NPTS
RETURN
Increase TIME by DT
K = I mod 40
Is K = 1 ?
no
yes
Skip to a new page and write headings
Write TIME and outputs at this time

ATAN3


ENTER
Is sine function = 0 ?
yes
X = Y
Y = π/2
RETURN
Left or right half plane ?
Left
Right
RH = -1.0
RH = 1.0
Is cosine function = 0 ?
yes
RETURN
no
Upper or lower half plane ?
Lower
Upper
UH = -1.0
UH = 1.0
Z ← RH √(X² + Y²)
Y ← RH UH × tan⁻¹ Y/X
X ← Z
RETURN